

Themengebiete „Künstliche Intelligenz“

- **Grundlagen**
 - Definition KI
 - TuringTest
 - KI Entwicklung
 - Agenten
 - Umgebungseigenschaften
 - Problemformulierung → Isomorphe Probleme
 - Problemtypen → Einzustands-, Mehrzustands-, Kontingenz-, Explorationsproblem
- **Suchen**
 - Vollständigkeit, Zeitkomplexität, Platzkomplexität, Optimalität
 - Uninformierte/Blinde Suche vs. Informierte/Heuristische Suche
 - Breitensuche
 - Uniforme Kostensuche
 - Tiefensuche
 - Tiefenbeschränkte Suche
 - IDS
 - Bidirektionale Suche
 - Suchstrategien bei CSPs
 - Backtracking vs. Vorwärtssuche
 - Bestensuche
 - Greedy Suche
 - Heuristiken
 - A* Suche
 - IDA*
 - Heuristiken für CSPs
 - Genetische Algorithmen
 - Brettspiele
 - Minimax Algorithmus
- **Wissensrepräsentation**
 - KB
 - Ebenen der Wissensrepräsentation
 - Aussagenlogik
 - KNF/DNF
 - Resolution
 - Davis Putnam Methode
 - Lokale Suche
 - GSAT
 - GWSAT
 - Prädikatenlogik
 - PNF
 - SNF
 - Herbranduniversum
 - Herbrandexpansion → Satz von Herbrand
 - Wünsche an PL2
 - Typen von Agenten
 - Situationskalkül von McCarthy
 - Positive-/Negative Effektaxiome
 - Rahmenproblem
 - Kausale Regeln
- **Handlungsplanung**
 - Handlungsplanung vs. Problemlösen
 - Strips Formalismus
 - Partiiell geordnete Pläne
 - Kausalitätsbedrohung → Demotion vs. Promotion
 - POP Algorithmus
 - Interferenz
 - Planungsgraph/Planungsproblem
 - Exklusivität von Fakten und Aktionen
 - Rückwärtssuche

- Problemlösen unter Unsicherheit
- **Wahrscheinlichkeitsrechnung**
 - A Priori / A Posteriori Wahrscheinlichkeit
 - Wahrscheinlichkeitsvektor
 - Die 4 Axiome der Wahrscheinlichkeitstheorie
 - Weltmodell und Theorem von de Finetti
 - Verbundwahrscheinlichkeiten
 - Bayessche Regel
 - Normalisierung
 - Multiple Evidenzen
 - Bayessche Netze
- **Nutzentheorie**
 - MEU-Prinzip
 - Axiome der Nutzetheorie → Orderability, Transitivität, Kontinuität, Substituierbarkeit, Monotonie, Dekomponierbarkeit
 - Deterministische vs. Stochastische Entscheidung
 - Markov Entscheidungsproblem
- **Lernen**
 - Wie lernt ein Agent?
 - Rückkopplung
 - Induktives Lernen
 - Entscheidungsbäume
 - Ockham's Razor
 - Rekursives Lernverfahren
 - Trainingsmenge vs. Testmenge
 - Bewertungen
 - Informationsgewinn
 - Current Best-Hypothesis
 - Version-Space/Candidate Elimination
 - Boundary Sets
 - Pac Lernen
 - Lernen von Entscheidungslisten
 - Neuronale Netze
 - Künstliche Neuronale Netze
 - Aktivierungsfunktion
 - Netzwerktopologieklassen
 - Rekurrente Netze
 - Hopfield Netze
 - Boltzmann Maschinen
 - Feed-Forward Netze
 - Geschichtete Feed Forward Netze (GFF)
 - Lernen im Neuronalen Netz
 - Optimal Brain Damage Heuristik
 - Weitere Netzwerktopologieklassen
 - Perzeptron
 - Perzeptron Lernregel
 - Mehrschichtige FF Netze
 - Backpropagation

• Fragenkatalog Künstliche Intelligenz

1. **Was ist eigentlich Künstliche Intelligenz?** [Der Versuch Computer „intelligenter“ zu machen |Die Natur der menschlichen Intelligenz besser zu verstehen]
2. **Welche 4 Vorgehensweisen gibt es den Begriff einzugrenzen?** [Definition über das Denken|Definition über das Handeln|Definition am menschlichen Vorbild|Ideale/Normative Definition]
3. **Was versucht man mit dem Turing Test herauszufinden?** [Vergleich Mensch vs. Maschine → Wenn es unmöglich ist zu unterscheiden ob das Gegenüber Mensch oder Maschine ist, dann hat die Maschine den Turing Test bestanden]
4. **Welche Anwendungsfelder der KI kennen Sie?** [Sprachverstehende und –generierende Systeme|Bildverstehende Systeme|Robotik|Assistenzsysteme]
5. **Welche Methoden der KI kennen Sie?** [Problemlösen und Suche|Wissensrepräsentation und –verarbeitung|Handlungsplanung| Maschinelles Lernen| Verarbeitung unsicheren Wissens| Neuronale Netze]
6. **Wie wurde die anfängliche Euphorie der KI in den 70er Jahren deutlich gedämpft?** [Anfänglich (ab 1956) ging man davon aus, dass sich die KI entwicklung genauso schnell wie die Computerentwicklung fortsetzen würde und das schon in „naher Zukunft“ Maschinen die gleichen Probleme lösen können wie der Mensch. Vorallem in den 60er Jahren konnte man viele Probleme auch in Mikrowelten lösen, das Probleme war dann aber, dass diese Modelle sich nicht auf die reale Welt skalieren liessen. Desweiteren benötigt intelligentes Verhalten viel Wissen, man geht also eher über zu Wissensbasierten Systemen]
7. **in den 80ern gab es dann doch noch eine Wende warum?** [Besonders in den 80er Jahren großer Erfolg mit Expertensystem (Wissensbasiert) → Erneute Flaute → KI Winter Ende der 80er Jahre]
8. **Die Moderne KI beginnt Mitte der 90er Jahre, welcher Paradigmenwechsel hat hier stattgefunden?** [Einzug probabilistischer Methoden, agentensichtweise, Formalisierung und Mathematisierung von KI-Techniken → Die moderne KI ist eine Mischung aus vielen koexistierenden Paradigmen]
9. **Was ist ein Agent?** [Ein Agent nimmt seine Umwelt durch Sensoren wahr (perzepte) und kann diese mit Aktoren Manipulieren (Aktionen)→ z.B. Menschen, Tiere, Roboter, Softbots, Heizungen, ABS]
 - a. **Was ist nun aber ein rationaler Agent?** [Macht immer „Das Richtige“. Leistungskriterien z.B. in m²/h oder Stromverbrauch beim Staubsauger etc.]
 - b. **Warum ist Optimales Verhalten oft unmöglich?** [Probleme: nicht alle relevanten Informationen wahrnehmbar; Berechnung zu komplex]
 - c. **Was ist der Unterschied zwischen einem allwissenden Agenten und einem rationalen?** [Allwissender: Kennt die tatsächlichen Effekte seiner Aktionen| rationaler: Handlung aufgrund seiner Wahrnehmungen und seines Wissens und versucht die erwartete Leistung zu maximieren]
 - d. **Was versteht man unter einem idealen, rationalen Agenten?** [Agent, der für alle möglichen Wahrnehmungssequenzen und gegebenem Weltwissen die Aktion wählt, die die Leistung maximiert → Wahrnehmungssequenz X Weltwissen → Aktion]
 - e. **Kennen Sie Beispiele für rationale Agenten?** Nennen Sie mir mal einen? [Interaktiver Englisch Tutor: Perzept: Getippte Wörter| Aktionen: Übungsaufgaben ausgeben, Vorschläge machen und Verbesserungen machen| Ziele: Notenergebnis des Lernenden zu maximieren| Umgebung: Mehrere Studenten]
10. **Welche Eigenschaften kann die Umgebung eines rationalen Agenten haben?** [zugänglich vs. Unzugänglich|deterministisch vs. Indeterministisch|episodisch vs nichtepisodisch(zB Schachspielen)→Kann die Qualität einer Episode bewertet werden oder ist die zukünftige Entwicklung für die Qualitätsbewertung ausschlaggebend|Statisch vs. Dynamisch|Diskret vs. Kontinuierlich]
 - a. **Welche der Kriterien sind intuitiv besonders schwierig zu bearbeiten?** [Unzugängliche, nichtepisodische, dynamische, kontinuierliche Umgebungen]
11. **Was ist ein problemlösender (zielorientierter) Agent?** [Er hat ein klar formuliertes Problem und Ziel| Der Anfangszustand ist fest definiert| Gewünscht: erreichen eines bestimmten Ziels (eines Zustandes) durch Ausführen geeigneter Aktionen]
12. **Die Art und Weise wie ein Problem formuliert ist, ist ausschlaggebend auf die Schwierigkeit der Lösung. Nennen Sie ein Beispiel für isomorphe Problembeschreibungen unterschiedlicher Lösungsschwierigkeit!** [Schachbrett mit

- abgeschnittenen Ecken, einmal schwarze und weiss Felder eingezeichnet und einmal ganz weiss| Mehrere Isomorphe Darstellungen von TIC TAC TOE]
13. **Welche Problemtypen über Wissen und Zustände gibt es?**
[Einzustandsproblem|Mehrzustandsproblem|Kontingenztproblem| Explorationsproblem]
 - a. **Was versteht man unter dem Einzustandsproblem?** [Vollständiges Welt- und Aktionswissen→ Agent weiß immer genau in welchem Weltzustand er sich befindet]
 - b. **Was versteht man unter dem Mehrzustandsproblem?** [unvollständiges Weltzustands- oder Aktionswissen→Der Agent kennt nur eine Menge möglicher Zustände. In welchem er genau ist, weiß er nicht| Kennt die Anzahl der Weltzustände dann ist die Anzahl der Zustände für die Suche entspricht der Potenzmenge über die Anzahl der möglichen Weltzustände]
 - c. **Was versteht man unter dem Kontingenztproblem?** [Der Agent kann keine komplette Lösungssequenz berechnen, da er zwar den Zielzustand kennt, die Zwischenzustände aber nicht alle]
 - d. **Was versteht man unter dem Explorationsproblem?** [Der Zustandsraum und die Zustandsübergänge (d.h. die Effekte der Aktionen) sind nicht vollständig bekannt]
 14. **Erklären Sie die folgenden Begriffe:**
 - a. **Anfangszustand?** [Zustand, von dem der Agent glaubt, anfangs zu sein]
 - b. **Zustandsraum?** [Menge aller möglichen Zustände]
 - c. **Operator?** [Beschreibung einer Aktion durch Angabe des resultierenden Zustands]
 - d. **Zieltest?** [Test, ob die Beschreibung eines Zustands einem Zielzustand entspricht]
 - e. **Pfad?** [Sequenz von Aktionen, die von einem Zustand in einen anderen führen.]
 - f. **Pfadkosten?** [Kostenfunktion g über Pfaden. Setzt sich üblicherweise aus der Summe der Kosten der Aktionen zusammen.]
 - g. **Lösung?** [Pfad von einem Anfangs- zu einem Zielzustand]
 - h. **Suchkosten?** [Zeit- und Speicherbedarf, um eine Lösung zu finden]
 - i. **Gesamtkosten?** [Suchkosten+Pfadkosten]
 15. **Erkläre die obigen Begriffe anhand des 8er-Puzzles, 8 Damen Problems bzw. des Missionar und Kannibalen Problems!**
 16. **Nennen Sie einige Beispiele für reale Probleme!** [Routenplanung, Finden kürzester Wege| Planung von Rundreisen (TSP)| VLSI Layout| Roboter Navigation (mit viele Freiheitsgraden)| Montageplanung]
 17. **Wie kann man allgemein nach einer Lösung suchen?** [mittels einem Suchbaum. Man exploriert alle Nachfolgezustände eines Knotens und testet ob man im Zielzustand ist]
 - a. **Wie sieht die Datenstruktur eines Knotens im Suchbaum aus?** [State: Zustand des Zustandsraums| Parent-Node: Vorgängerknoten| Operator: Operator der den aktuellen Knoten erzeugt hat| Depth: Tiefe im Suchbaum| Path-Cost: Pfadkosten bis zu diesem Knoten]
 - b. **Was wird in der Warteschlange (Queue) abgespeichert?** [noch nicht explorierte Knoten. Wird die Queue leer (keine Expansion mehr möglich), und man hat keinen Zielzustand erreicht, gibt es keine Lösung]
 18. **Welche Kriterien gibt es für unterschiedliche Suchstrategien?** [Vollständigkeit: Wird immer eine Lösung gefunden, sofern es eine gibt?| Zeitkomplexität: Wie lange dauert es (im schlechtesten Fall), bis eine Lösung gefunden ist| Platzkomplexität: Wieviel Speicher benötigt die Suche (im schlechtesten Fall)?| Optimalität: Findet das Verfahren immer die beste Lösung]
 19. **Welche Klassen von Suchstrategien gibt es?** [uninformierte/blinde Suche: keine Information über Länge/Kosten eines Lösungspfades| informierte/heuristische Suche]
 - a. **Geben sie Beispiele für uninformierte Suche** [Breitensuche, uniforme Kostensuche, Tiefensuche| tiefenbeschränkte Suche, iterative Tiefensuche| bidirektionale Suche]
 20. **Wie funktioniert Breitensuche?**
 - a. Was ist der große Nachteil dieser Suchstrategie? [sehr hohe Kosten]
 - b. Wann ist die Lösung optimal? [Wenn die Pfadkosten eine nichtfallende Funktion der Knotentiefe ist]
 21. **Wie funktioniert die uniforme Kostensuche?** [Es wird immer der Knoten mit geringsten Pfadkosten expandiert (ähnlich zu Breitensuche)]
 - a. Unter welchen Bedingung findet diese Strategie immer die günstigste Lösung? [Wenn die Pfadkosten des Nachfolgeknotens immer größer sind als die des Vorgängers]
 22. **Wie funktioniert Tiefensuche?** [Expandiere immer einen nicht expandierten Knoten mit maximaler Tiefe]

23. **Was ist der Unterschied der tiefenbeschränkten Suche zur normalen Tiefensuche?** [Die Tiefe wird beschränkt]
 a. **Was ist das Problem?** [maximale Tiefe muss den Durchmesser des Problems entsprechen, sonst wird die Lösung nicht gefunden]
24. **Aus welcher Kombination von Suchstrategien entsteht die iterative Tiefensuche?** [aus Tiefensuche und Breitensuche] die Tiefe der Tiefensuche wird iterativ inkrementiert
 a. **Welchen Vorteil hat diese Strategie gegenüber der Breitensuche?** [weniger Speicherplatz]
 b. **Was ist die Zeit- und Platzkomplexität wenn b der Verzweigungsgrad, und d die Tiefe ist?** [Zeit: $O(b^d)$ Platz: $O(b \times d)$]
 c. **Wann wird die iterative Tiefensuche bevorzugt verwendet?** [bei großen Suchräumen mit unbekannter maximaler Suchtiefe]
25. **Wie funktioniert die bidirektionale Suche?** [man startet gleichzeitig beim Start- und beim Zielzustand, und hofft das man irgendwann mal aufeinander trifft]
 a. **Welche Suchzeit erwartet man, wenn die Suche in beiden Richtungen symmetrisch ist?** [$O(2 \cdot b^{d/2}) = O(b^{d/2})$]
 b. **Welche Probleme ergeben sich bei bidirektionaler Suche?** [Operatoren nicht immer umkehrbar|Viele Zielzustände (z.B. Was ist der Vorgänger des „Schachmatt“)] Es ist schwer herauszufinden ob sich die beiden Suchen getroffen haben| Welche Art von Suche wählt man für die beiden Richtungen?]
26. **Vergleichen Sie die verschiedenen Strategien anhand der Zeit-/Platzkomplexität, Optimalität und Vollständigkeit!**

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Time	b^d	b^d	b^m	b^l	b^d	$b^{d/2}$
Space	b^d	b^d	bm	bl	bd	$b^{d/2}$
Optimal?	Yes	Yes	No	No	Yes	Yes
Complete?	Yes	Yes	No	Yes, if $l \geq d$	Yes	Yes

27. **Was ist die Klasse der Constraint-Satisfaction Probleme (CSP)?** [Zustände durch Werte von Variablen definiert] Operatoren belegen eine Variable mit einem Wert| Der Zieltest wird durch Constraints spezifiziert, welche die Variablenbelegung erfüllen müssen| Der Zielzustand ist dann eine Belegung der Variablen mit Werten, die alle Constraints erfüllen]
28. **Erläutern Sie das CSP anhand der 8-Damen Problem oder dem Einfärben von Graphen!** [8 Damen: 8 Variablen $V_1 \dots V_8$ wobei i die Spaltenposition ist| Zwischen den Paaren von Variablen gibt es nun Constraints, die die Nichtangreifbarkeit ausdrücken]| Einfärben von Graphen: Zu Jedem Knoten V_i werden wieder Constraintbedingungen formuliert]
29. **Welche Besonderheiten ergeben sich beim Lösen des CSP?** [Tiefensuche ist bei CSPs vollständig] Verzweigungsfaktor ist u.U. sehr hoch| Reihenfolge der Variableninstantiierung unerheblich für die Lösung| Nach jeder Operatoranwendung kann geprüft werden, ob Constraints verletzt wurden → Falls ja folgt Backtracking]
30. **Backtracking stellt bei der Suche u.U. erst im Blatt fest, dass es der falsche Weg ist, d.h. der gesamte Unterbaum wurde umsonst durchsucht, wie kann man das verhindern?** [Vorwärtstest: Bei allen nicht belegten Variablen werden die Werte eliminiert (d.h. Wertebereich eingeschränkt), die nicht mehr möglich sind. Backtracking muss nun erst dann angewandt werden, wenn der Wertebereich (Domain) einer Variablen leer ist| Bei einer noch spezielleren Version wird in jedem Schritt auf Kantenkonsistenz geachtet, d.h. der Wertebereich einer Variablen darf immer nur Werte enthalten die nicht im Widerspruch zu den Variablen-Constraints stehen]
31. **Welches ist der Unterschied zwischen informierter Suche und uninformatierter?** [Uninformierte: Keine Informationen über Kosten von aktuellem Knoten zum Zielknoten] Informierte Suche: Eine Evaluierungsfunktion schätzt die Kosten bis zum Ziel ab]
32. **Wie funktioniert Bestensuche?** [Knoten mit dem besten h Wert werden expandiert, wenn h bereits den tatsächlichen Entfernungswert ausgibt braucht man diesen nicht mehr zu suchen]
33. **Eine spezielle Variante der Bestensuche ist die so genannte Greedy Suche, welche Eigenschaften hat diese?** [h Funktion: Schätzt den Abstand zum Ziel (z.B. Luftlinienentfernung, ausserdem $h(n)=0$ falls n bereits der Zielknoten ist]

34. **Was versteht man unter einer Heuristik?** [Heuristische Verfahren wurden vom Mathematiker Poly eingeführt um Problemlösungstechniken zu beschreiben. Eine Heuristik ist demnach eine schnelle aber u.U. unvollständige Methode, um Probleme zu lösen, die die Sucher aber im Normalfall bechleunigt. Ein Heuristik ist problemspezifisch und fokussiert]
35. **Was ist die A* Suche?** [A* besteht aus zwei Funktionen, nämlich der Funktion $g(n)$ die die tatsächlichen Kosten meiner bereits zurückgelegten Strecke vom Anfang bis n beschreibt und der Funktion $h(n)$, die die noch zurückzulegende Strecke bis zum Ziel abschätzt ($h(n)$ ist eine Heuristik). Meine Geschätzten Kosten des günstigsten Pfades orientieren sich nun aber nicht nur an $h(n)$ wie bei Greedysuchen, sondern an $g(n)+h(n)=f(n)$.]
- Wann ist eine heuristische Funktion $h(n)$ zulässig?** [$h(n)$ ist zulässig, wenn alle Funktionswerte von $h(n)$ immer unter der tatsächlichen Entfernung (also den Funktionswerten von $h^*(n)$) liegen, als $h(n) \leq h^*(n) \rightarrow h(n) =$ Luftlinie wäre z.B. zulässig]
 - Was sind Konturen in A*?** [Knoten die innerhalb einer Kontur liegen, haben eine f -Wert, der kleiner ist als der Kontur-Wert]
 - Warum ist A* Suche immer optimal?** [Beweis durch Widerspruch: Annahme A* findet nicht den optimalen Zielknoten sondern einen anderen Knoten G_2 mit $g(G_2) > f^*$] n sei ein Knoten auf dem Weg von Start- zu Zielknoten $G \rightarrow f(n) \leq f^*$ | Da n nicht vor G_2 expandiert wurde, gilt $f(G_2) \leq f(n)$ (n liegt ja auch gar nicht auf dem Weg zu G_2) | Weiter gilt dann $f(G_2) \leq f^*$ | Wegen $h(G_2) = 0$ gilt bei G_2 dass $g(G_2) \leq f^*$ | \rightarrow Widerspruch!]
 - Warum ist A* vollständig?** [A* findet eine Lösung, wenn es eine gibt. Folgendes muss aber gelten: 1. Jeder Knoten hat nur endlich viele Nachfolgerknoten – 2. Konstante δ , sodass jeder Operator mindesten die Kosten δ hat]
 - Welche Komplexität hat A*?** [Normalerweise exponentiell! – Wenn aber gilt: $|h^*(n) - h(n)| \leq O(\log(h^*(n)))$ werden nur subexponentiell viel Knoten expandiert]
 - Nennen Sie anhand des Schiebepuzzles verschiedene heuristische Funktionen!** [zB h_1 : Zahl der Kästchen in falscher Position | h_2 Summe der Distanzen der Kästchen zu ihrer Zielposition (Manhattan Distanz)]
 - Es gibt auch eine Kombination aus A* und der iterativen Tiefensuche (IDS) namens IDA*, wie funktioniert das?** [Es werden alle Knoten innerhalb einer Kontur abgesucht \rightarrow Statt Tiefenbeschränkung erfolgt eine Kostenbeschränkung]
36. **Welche Heuristiken gibt es zur Lösung von Constraint Satisfaction Problems (CSPs)?** [Eingeschränkteste Variable zuerst | Einschränkendste Variable zuerst | Am wenigsten einschränkenden Wert zuerst]
37. **Was versteht man unter lokaler Suche mit Hill-Climbing?** [Start mit zufällig gewählter Konfiguration, sukzessive Verbesserung (Hill Climbing)]
- Welche Probleme ergeben sich insbesondere beim Hill-Climbing? [Lokale Maxima | Plateaus | Grate]
 - Welche Lösungs möglichkeiten gibt es dazu? [Neustarts, wenn keine Verbesserung mehr eintritt | Rauschen injizieren, d.h. randomisieren | Tabu-Suche, die letzten n angewandten Operatoren nicht anwenden]
38. **Was ist der Unterschied zwischen Hill Climbing und Simulated Annealing?** [Der Unterschied ist, dass Simulated Anealing im Prinzip ein Hill-Climbing mit permanentem abnehmen des Rauschens ist]
39. **Was ist das Qualitätsmaß einer zufälligen CSP Konfiguration?** [Anzahl der bereits erfüllten Constraints]
40. **Was versteht man in diesem Zusammenhang dann unter heuristischen reparieren?** [Lokale Suche nach Erfüllung der restlichen Constraints (Bsp.: Hubble-Teleskop)]
41. **Was versteht man unter einem genetischen Algorithmus?** [Wie bei der Evolution gute Lösungsansätze durch Kreuzung, Mutation und Selektion (mittels Fitnessfunktion) weiter verbessern. Siehe auch Algorithmentheorie]
42. **Warum wird bei der Erklärung von KI gerne auf bekannte Brettspiele zurückgegriffen?** [Brettspiele sind sehr abstrakt, erfordern eine gewisse Form von Intelligenz, Zustände sind einfach darstellbar, Aktionsmöglichkeiten sind wohldefiniert | Wir können die Lösung als ein Suchproblem realisieren, Die Weltzustände sind voll zugänglich, wir haben eine Kontingenzzproblem, weil wir die Züge des Gegners nicht voraussagen können]
43. **Nennen Sie Brettspiele, deren Suchbaum besonders groß und komplex ist!** [Schach 35^{100} Knoten | Go-Bang: 200^{300} Knoten]
44. **Welche Eigenschaften hat ein gutes „Spielprogramm“?** [Irrelevante Äste werden abgeschnitten | gute Evaluierungsfunktion | möglichst viele Halbzüge vorausschauen]
45. **Erkläre Minimax anhand folgender Begriffe: Spieler, Anfangszustand, Operatoren, Terminierungszustand, Nutzenfunktion, Strategie!!**

- a. **Wie sieht der Suchbaum z.B. bei Tic-Tac-Toe aus?** [Abwechselnd Ebenen für MIN und MAX| MAX beginnt]
- b. **Wann liefert der Minimax-Algorithmus eine optimale Strategie für MAX?** [wenn es möglich ist, den gesamten Suchbaum(Spielbaum) zu erzeugen]
- c. **Wie funktioniert der Minimax Algorithmus genau?** [1. Erzeuge vollständigen Spielbaum mit Tiefensuche| 2. Wende die Nutzenfunktion auf jeden Terminalzustand an| 3. Beginnend bei Terminalzuständen, berechne die Werte für die Vorgängerknoten wie folgt: Knoten ist MIN-Knoten: Wert ist das Minimum der Nachfolgerknoten, Knoten ist MAX-Knoten: Wert ist das Maximum der Nachfolgerknoten| MAX wählt dann im Anfangszustand den Nachfolgerknoten mit größtmöglichem Nutzen (Minimax-Entscheidung)]
- d. **Wovon geht der Minimax-Algorithmus bezüglich den Zügen von MIN aus?** [dass MIN perfekt spielt, Fehler von MIN können das Ergebnis für MAX nur verbessern]
- e. **Was muss getan werden, wenn der Spielbaum zu tief ist?** [Der Minimax-Wert kann nur noch approximiert werden]
- f. **Nenne Beispiele für einfache Bewertungskriterien beim Schach?** [Wert der Spielfiguren (Bauer<Dame)| Sicherheit des Königs, Positionierung der Bauern]
- g. **Welches Kriterium sollte die Evaluierungsfunktion erfüllen?** [monoton steigend im Bezug auf die Gewinnchance]
- h. **Bevorzugt werden „gewichtete lineare Funktionen“ als Evaluierungsfunktionen gewählt. Wie sehen diese aus?** [$w_1f_1 + w_2f_2 + \dots + w_nf_n$, wobei f_i das i-te Kriterium und w_i das i-te Gewicht bezeichnen]
- i. **Wie könnte der aktuelle Spielzustand beim Tic-Tac-Toe bewertet werden?** [Anzahl der noch möglichen vollständigen Zeilen für MAX minus Anzahl der noch möglichen vollständigen Zeilen für MIN]
- j. **Wie wird die untere Schranke (α -Wert) bzw. obere Schranke (β -Wert) berechnet?**
- k. **Wie hoch ist der Effizienzgewinn im besten und im durchschnittlichen Fall?** [Bester Fall: Wenn immer der beste Zug zuerst gemacht wird, reduziert sich der Suchaufwand auf $O(b^{d/2})$ | Durchschnittlicher Fall: Wenn die Züge zufällig verteilt sind, reduziert sich der Suchaufwand auf $O((b/\log b)^d)$]
 - i. **Wie hoch ist der Effizienzgewinn in der Praxis?** [Schon mit relativ einfachen Anordnungsheuristiken kommt man in die Nähe des besten Falls
→ wir können doppelt so tief in der gleichen Zeit suchen]
- l. **Was muss man bei Spielen mit Zufallskomponenten (Würfel) zusätzlich beachten?** [Einfügen von Würfelknoten in dem Spielbaum]
- m. **Aus welchen Komponenten setzt sich die Berechnung des erwarteten Nutzens des Würfelknotens C über MAX zusammen?** [d_i : mögliche Würfelresultate| $P(d_i)$: Die Wahrscheinlichkeit für ein bestimmtes Würfelresultat| $S(C, d_i)$: Von C erreichbare Positionen beim Wurf d_i | $utility(s)$: Bewertung von $s \in S(C, d_i)$]
 - i. **Wie sieht die Nutzenfunktion für MAX aus?**

$$expectimax(C) = \sum_i P(d_i) \max_{s \in S(C, d_i)} (utility(s))$$

- ii. **Welche Probleme ergeben sich mit Anordnungsinvarianten Transformationen von Evaluierungsfunktionen?** [Sie erhalten nicht die Ordnung zwischen den Zügen]
- iii. **Inwieweit steigen die Suchkosten, wenn n nun die Anzahl möglicher Würfelresultate ist?** [von $O(b^d)$ auf $O(b \times n^d)$]

46. **Bisher haben wir hauptsächlich von rational handelnden Agenten gesprochen- Rationales Handeln setzt aber oft rationales (logisches) Denken voraus. Was muss der Agent dazu auf jeden Fall Wissen?** [Er muss die Welt symbolisch in einer Wissensbasis (KB) repräsentieren]
 - a. **Aus was besteht die Wissensbasis?** [Sie enthält Sätze in einer Sprache mit Logik, d.h. wir können Sätze als Aussagen interpretieren (Semantik). Die Sätze haben alleine durch ihre Form (Syntax) einen kausalen Einfluss auf das Verhalten des Agenten in einer Weise, die in Korrelation zum Inhalt der Sätze steht]
 - b. **Wie interagiert der Agent mit der KB?** [$ASK(KB, \alpha) = JA$ bzw. $TELL(KB, \alpha) = KB'$ bzw. $FORGET(KB, \alpha) = KB'$]

47. **Welch 3 Ebenen der Wissensrepräsentation unterscheidet man?**
 [Wissensebene(Abstrakteste Ebene)→Umfasst das gesamte Wissen der KB|Symbolische Ebene→Kodierung in formaler Sprache|Implementierungsebene→Interne Darstellung der Sätze zB als String oder Wert in einer Matrix]
- a. **Unter welchen Umständen reicht es sich auf der Wissensebene zu bewegen?**
 [Wenn ASK und TELL korrekt funktionieren]
- i. **Welche Vorteile hat man hierdurch?** [Man hat eine sehr komfortable Benutzerschnittstelle]
48. **In wiefern kann ein wissensbasierter Agent seine Wissensbasis nutzen?** [Er kann mit ihr sein Hintergrundwissen repräsentieren, seine Beobachtungen speichern, seine durchgeführten Aktionen speichern, und seine Handlungen aus ihr ableiten]
49. **Was ist die Wumpus Welt?** [Einfache Repräsentation eines Problems die ein Agent durch Einträge in seiner Wissensbasis lösen kann]
50. **Wie kann man Wissen ausdrücken?** [Mit Hilfe einer deklarative und präzisen Sprache wie zB der Aussagenlogik]
- a. **Welches sind die Grundbausteine der Aussagenlogik?** [Atomare, nicht mehr weiter zerlegbare Aussagen|Logische Verbindungen wie und, oder, nicht zum **Aufbauen von Formeln**]
- b. **Für was brauchen wir das?** [Wir wollen wissen, wann eine Aussage wahr ist und wann eine Aussage aus einer KB folgt| Wir wollen Ableitungen aus vorhandenen Aussagen konstruieren können]
- c. **Welche Aussagen kann eine Formel ϕ ergeben, wenn sie mit Variablen der KB I gefüllt wird?** [Erfüllbar|Unerfüllbar|falsifizierbar|Tautologie]
- d. **Wann sind zwei Formeln logisch äquivalent?** [Wenn für alle I aus der KB gilt, dass die beiden Formeln mit genau den Belegungen I immer gleiche Ergebnisse haben]
- e. **Wie kann man u.A. feststellen, ob eine Formel unerfüllbar, erfüllbar etc. ist?**
 [Wahrheitstabelle aufstellen]
51. **Was ist eine Konjunktive-/Disjunktive Normalform?** [KNF: Konjunktion von disjunktionen von Literalen ((a oder b) und (c oder d))| DNF: Disjunktion von Konjunktion von Literalen ((a und b) oder (c und d))|Jede Formel ist in KNF bzw. DNF überführbar!]
- a. **Wie erzeugt man eine KNF?** [Eliminiere \Rightarrow und \Leftrightarrow | Schiebe „nicht“ nach innen|Verteile „oder“ über „und“|Vereinfache (a „oder“ a = a)]
- b. **Wie erzeugt man eine DNF?** [Sehr einfach über Wahrheitstabelle! Andere Verfahren möglich]
- c. **Was besagt der Deduktionssatz?**

$$[KB \cup \{ \phi \} \models \psi \text{ gdw. } KB \models \phi \Rightarrow \psi]$$
- d. **Was besagt der Kontrapositionssatz?**

$$[KB \cup \{ \phi \} \models \neg \psi \text{ gdw. } KB \cup \{ \psi \} \models \neg \phi]$$
- e. **Was besagt der Widerspruchssatz?**

$$[KB \cup \{ \phi \} \text{ ist unerfüllbar gdw. } KB \models \neg \phi]$$
- f. **Können Sie die Sätze Beweisen?** [öhhh...ja...]
- g. **Wann heisst ein Kalkül korrekt?** [Wenn alle aus einer KB ableitbaren Formeln auch tatsächlich logisch folgen. Anders ausgedrückt, bedeutet korrekt, dass keine erfüllbare Formelmenge durch den Kalkül als vermeintlich unerfüllbar nachgewiesen werden kann]
- h. **Wann heisst ein Kalkül vollständig?** [Falls jede Formel, die logisch aus KB folgt, auch aus KB ableitbar ist. Anders ausgedrückt, bedeutet Vollständigkeit umgekehrt, das jede unerfüllbare Formelmenge durch den Kalkül als solche nachgewiesen werden kann]
52. **Was ist die Idee von Resolution?** [Ableitungsverfahren, welches nicht alle Interpretationen durchprobiert→ Man versucht zu zeigen, dass eine Formelmenge unerfüllbar ist| Formeln müssen in KNF sein|Worst Case ist trotzdem Exponentiell]
- a. **Wie lautet die Resolutionsregel und erklären Sie diese!**

$$\frac{C_1 \cup \{l\}, C_2 \cup \{\bar{l}\}}{C_1 \cup C_2}$$

 [Wenn wir je zwei Mengen von Literalen haben, welche ein bestimmtes Literal einmal negiert und einmal nichtnegiert enthalten, so kann man

dieses Literal „herauskürzen“ – Die entstehende Resolvente ist aber nicht mehr äquivalent zu den Elternklauseln, sondern folgt nur aus ihnen]

- b. **Was besagt die Notation $R(\Delta)$?** [Die Ursprüngliche Klauselmenge erweitert um die neu entstandenen Resolventen zweier Klauseln aus Δ , also $\Delta \cup \{C|C \text{ ist Resolvente zweier Klauseln aus } \Delta\}$]
 - c. **Was besagt die Notation $\Delta \vdash D$?** [Die ist eine Menge verschiedener Resolventen, die aus Δ bzw. aus $\Delta \cup$ mit allen Resolventen der vorhergehenden Schritte hervorgegangen sind. Also, wenn $D = C_1, C_2, \dots, C_n$ ist, dann:

$$C_i \in R(\Delta \cup \{C_1, \dots, C_{i-1}\}), \text{ für } 1 \leq i \leq n$$
]
 - d. **Was muss gelten, damit Resolution sowohl Korrekt?** [Für die Korrektheit muss gelten $\Delta \vdash D$, dann $\Delta \vdash D$]
 - e. **Was muss gelten damit Resolution vollständig ist?** [Für die Vollständigkeit müsste genau das umgekehrte auch gelten, also $\Delta \not\vdash \varphi$ impliziert $\Delta \vdash \varphi$. Gilt aber leider nicht!!! Resolution ist nur Widerlegungsvollständig d.h. Δ unerfüllbar impliziert $\Delta \vdash \square$]
 - f. **Was können wir also mit Hilfe des Widerspruchstheorems zeigen?** [Dass $KB \vdash \varphi$]
53. **Warum ist die Aussagenlogik etwas umständlich (gerade für die Wumpus Welt)?** [Weil man für jedes zu betretende Feld eine eigene Regel aufbauen muss. Ausserdem gelten die Regeln eigentlich nur Statisch oder müssten regelmässig neu aufgebaut werden, da sich der Wumpus ja eigentlich bewegen kann! → Mächtigere Logik wie Prädikatenlogik nötig !]
54. **Wie funktioniert das Verfahren von Davis-Putnam?** [Der Algorithmus ist rekursiv! 1. Wenn die Klauselmenge Δ leer ist, dann gib „erfüllbar“ aus| 2. Wenn $\square \in \Delta$, dann gib „unerfüllbar“ aus| 3. „Unit-Propagation-Rule“: Wenn Δ eine Einheitsklausel (Klausel mit nur einm Literal) enthält, belege diese so, dass sie erfüllt ist!| 3. Split-Regel: Nimm eine noch nicht belegte Variable aus der Variablenmenge und belege sie mit „true“, rufe DP rekursiv auf, liefert dieser Aufruf am Ende eine unerfüllbar, dann belege die Variabel mit „false“ statt mit „true“ rufe nun erneut DP rekursiv auf]
55. **Nennen Sie einige Eigenschaften von DP!** [DP ist exponentiell → Split-Regel| DP ist polynomiell, wenn er auf Hornklauseln angewandt wird (Hornklauseln haben nur ein positives Literal)|DP ist vollständig, korrekt und terminierend|DP konstruiert ein Modell falls eines vorhanden ist!|Gute Heuristiken zur Variablenauswahl werden benötigt]
56. **Erfüllbarkeitsprobleme sind nicht in allen Bereichen gleich schwer zu lösen. Wie unterscheiden sich diese „Problemspitzen“?** [Die Erfüllbarkeitsprobleme haben so genannte Phasenübergängen, d.h. z.B. bei 3-Sat einen Bereich, in dem die Wahrscheinlichkeit zur Erfüllung fast 100% ist (Under-Constrained), ab einem bestimmten Verhältnis von Anzahl der Klauseln durch die Anzahl der enthaltenen Variablen in der Formel schwenkt diese Eigenschaft radikal um ins schlechtere (bei 3-Sat bei 4.3 Klauseln)(Overconstrained Bereich). Die Davis Putnam Methode hat ausgerechnet an diesen Phasenübergängen enorme Laufzeiten!| Bei gleich bleibendem Klausel/Variabel Verhältnis steigt die Laufzeit bei Sukzessiver Erhöhung beider Mengen stark an| Lässt man dagegen die Anzahl der Klauseln konstant und erhöht nur die Variablen Anzahl, dann bleibt die Laufzeit nahezu unverändert]
- a. **Was bedeutet eigentlich das k bei einem k-SAT Problem?** [Die maximale Anzahl der Variablen pro Klausel]
57. **Neben der DP Methode gibt es auch eine leider unvollständige Methode, die aber trotzdem viele sehr große Instanzen lösen kann, wie heißt denn diese?** [Lokale Suche: Versucht, ausgehend von einer zufälligen Konfiguration, durch lokale Modifikation bessere Konfigurationen zur erzeugen]
- a. **Was ist bei der lokalen Suche das Hauptproblem?** [Das Hängenbleiben in lokalen Extrema obwohl man eigentlich globale Extrema finden (muss) bei denene alle Constraints/Klauseln erfüllt sind]
 - b. **Wie kann man nun aber die Qualität (Gütemaß) einer Konfiguration bewerten?** [z.B. bei logischen Problemen die Anzahl der erfüllten Constraints pro Klausel]
 - c. **Wie kann man aus lokalen Extrema „herauskommen“?** [Einbringen von Rauschen und/oder Neustarts]
 - d. **Warum wird bei all den Unwägbarkeiten dir lokale Suche dennoch angewandt?** [Sie liefert vor allem bei KNF Formeln erfüllbare Belegungen selbst ohne das Einbringen von Rauschen]
58. **Was ist GSAT?** [Ein weitere Strategie um Klauselmengen auf Erfüllbarkeit zu testen]
- a. **Was ist bei GSAT anders als bei der herkömmlichen lokalen Suche?** [Hill-Climbing nicht immer erforderlich, man kann sich auch auf einem Plateau bewegen (Seitwärtsbewegung)]

59. **Wie kann man GSAT weiter verbessern um nicht in lokalen Plateaus fest zu hängen?** [Simuliertes Abkühlen] GWSAT: Random Walk, d.h. wähle eine Variable in einer noch nicht erfüllten Klausel und ändere Ihrer Belegung mit Wahrscheinlichkeit p ansonsten wähle GSAT Strategie mit $(1-p)$ Random Noise: Wie Random Walk aber es wird irgendeine Klausel gewählt (evtl auch eine schon erfüllte)]
60. **Welches Einsatzgebiet hat der GSAT Algorithmus und dessen Varianten?** [Anwendungen mit sehr großen Problem instanzen, bei denen es sich lohnt, die Parameter anzupassen]
61. **Was ist der Unterschied zwischen Aussagenlogik und Prädikatenlogik?** [Neue Operatoren: „für Alle“ und „Existiert“|Funktionssymbole zur Repräsentation von Objekten(Gewicht/Farbe etc.)|Prädikatensymbole für Aussagen über Objekte: zB (Block,rot)]
62. **Welche Besonderheit hat ein 0-Stelliges Prädikat bzw eine 0-stellige Funktion in der Prädikatenlogik?** [Prädikat: wie aussagenlogische Atome|Funktion: Konstanten]
63. **Was ist ein Term?** [Ein Term ist ein Objekt-Eine Funktion ist auch ein Term: Bsp: Wer lernt mit ___? Antwort ist ein Objekt und kein Wahrheitswert! Jede Variable und Konstante ist ein Term] Wenn t_1, \dots, t_2 Terme dann auch $f(t_1, \dots, t_n)$
- a. Wann nennt man einen Term einen Grundterm? [Wenn er keine Variablen besitzt]
64. **Was sind atomare Formeln?** [Prädikate ohne aussagenlogische Verknüpfungen z.B. Block(x) liefert True, wenn x ein Block ist] Wenn t_1, \dots, t_n Terme sind und P ein n-stelliges Prädikat, dann ist $P(t_1, \dots, t_n)$ eine atomare Formel| Wenn t_1 und t_2 Terme sind, dann ist $t_1=t_2$ auch ein atomare Formel]
- a. **Was sind Grundatome?** [Atomare Formeln ohne Variablen]
65. **Was ist ein Prädikat?** [Im Prinzip ein natürlichsprachlicher Satz mit „Leersetellen“ d.h. Informationen die weggelassen werden, die Parameter sind die Auffüllungen für die Leerstellen: z.B. „ t_1 ist ein t_2 und wohnt in t_3 “ ist ein Prädikat mit 3 Parametern]
66. **Was ist eine Formel?** [Zusätzlich zur atomaren Formel, dürfen diese nun mit aussagen- und prädikatenlogischen Verknüpfungen verknüpft werden]
67. **Erklären Sie die Bedeutung folgender PL1 Formel! $\forall x[\text{Block}(x) \Rightarrow \text{Rot}(x)], \text{Block}(a)$?** [Für alle Objekte x gilt: Falls x ein Block ist, dann ist x rot. a ist ein Block.]
68. **Was ist eine Interpretation?** [Eine Instanz, also eine Belegung]
69. **Was sind freie bzw. gebundene Variablen?** [Frei: Wenn nicht an einen Quantor gebunden|Gebunden: Wenn an Quantor gebunden]
70. **Was sind geschlossene Formeln bzw. Sätze?** [Formeln in denen keine freien Variablen vorkommen]
71. **Was ist ein Modell?** [Eine Interpretation I heisst Modell von φ unter α , wenn $I, \alpha \models \varphi$]
72. **Wann sind zwei Formeln logisch äquivalent (φ und ψ)?** [Wenn für alle I, α gilt: $I, \alpha \models \varphi$ gdw. $I, \alpha \models \psi$]
73. **Was ist die Pränexe Normalform?** [Alle Quantoren am Anfang]
- a. **Wie bildet man die Pränexe Normalform (PNF)?** [1. Eliminieren von \Rightarrow und \Leftrightarrow 2. \neg nach innen ziehen] 3. Quantoren nach außen ziehen(evtl. Variablen umbenennen)
- b. **Gibt es einen Algorithmus der die PNF aus jeder Formel berechnet?** [Ja, wurde aber nicht behandelt]
74. **Was bringt uns die PNF?** [Leider ist die Bestimmung auf Erfüllbarkeit oder Allgemeingültigkeit nicht so einfach wie bei der Aussagenlogik, deshalb könnte man das Erfüllbarkeitsproblem der Prädikatenlogik auf die Erfüllbarkeit in Aussagenlogik reduzieren \rightarrow Die Menge der aussagenlogischen Formeln wird dann allerdings riesig!!! \rightarrow Aber jetzt kann man ganz normale Resolution anwenden]
75. **Was ist die Skolem-Normalform?** [PNF und alle Existenzquatoren eliminiert]
- a. **Wie geht das mit dem Eliminieren?** [$\varphi = \forall x_1 \dots \forall x_i \exists y \psi,$ wird zu $\varphi' = \forall x_1 \dots \forall x_i \psi[y/g_i(x_1, \dots, x_i)]$ d.h. die Variable y wird durch eine Funktion g_i ersetzt]
76. **Zu jeder geschlossenen Formen (d.h. ohne freie Variablen) kann man ja die SNF recht effektiv berechnen, was muss man dabei beachten?** [Keine Äquivalenztransformation, nur die Erfüllbarkeit wird erhalten] Sie ist nicht eindeutig!]
77. **Was versteht man unter dem Herbranduniversum (Grundtermmenge) über einer Menge von SNF Formeln Θ^* ?** [Die (abzählbare) Menge aller Grundterme, die sich mit

Symbolen aus Θ^* bilden lassen, falls es kein Konstantensymbol gibt, wird eines hinzugefügt]

78. **Was ist die Herbrandexpansion $E(\Theta^*)$?** [$E(\Theta^*)$ ist die Instanziierung der Matrizen (Restformel ohne Quantoren) aller Formeln in Θ^* durch alle Terme $t \in D(\Theta^*)$]

$$E(\Theta^*) = \{\psi_i[x_1/t_1, \dots, x_n/t_n] \mid (\forall x_1, \dots, x_n \psi_i) \in \Theta^*, t_j \in D(\Theta^*)\}$$

79. **Und was besagt nun der Satz von Herbrand?** [Sei Θ^* eine Menge von Formeln in SNF. Dann ist Θ^* erfüllbar gdw. $E(\Theta^*)$ ist. – Bem.: Wenn $D(\Theta^*)$ und Θ^* endlich sind, dann ist auch $E(\Theta^*)$ endlich!]
80. **Was macht man, wenn die Herbrandexpansion unendlich ist?** [Die Aussagenlogik ist kompakt, das heisst, dass jede (höchstens abzählbare) Menge von Formeln der Aussagenlogik erfüllbar ist, gdw. jede endliche Teilmenge erfüllbar ist → Wenn eine Teilmenge bereits unerfüllbar ist, dann ist es auch die Gesamte (unendliche bzw. höchstens abzählbare) Menge → Der erste Satz trifft auch auf die Prädikatenlogik zu !]
81. **Nehmen wir an, sie haben eine Formel, die Sie auf Allgemeingültigkeit überprüfen wollen, könnte man da nicht einfach in der Gesamtmenge der Allgemeingültigen Formeln nachsehen ob diese enthalten ist?** [Jein, man kann die Menge der allgemeingültigen (und auch die der unerfüllbaren) Formeln rekursiv aufzählen, dies ist aber nicht gerade effektiv, ausserdem kann man keine Zeitschranke angeben, nach welcher unser Verfahren mit Ergebnis terminiert. Ist die gesuchte Formel nicht allgemeingültig (bzw. unerfüllbar), so läuft das Verfahren unendlich → Es handelt sich um ein so genanntes Semientscheidungsverfahren]
82. **Und wie ist das mit den erfüllbaren Formeln?** Kann man diese überprüfen? [Leider ist es unentscheidbar (nur semi-Entscheidbar) ob eine Formel allgemeingültig ist, die Menge der erfüllbaren Formeln in PL1 ist auch nicht rekursiv aufzählbar, d.h. wir geraten bei erfüllbaren u.U. in eine Endlosschleife, bei der wir aber nicht sagen können ob Sie evtl. doch irgendwann terminiert mit dem Ergebnis die Formel sei allgemeingültig (bzw. unerfüllbar)]
83. **Welche Erweiterung würde man sich in einer PL2 wünschen?** [Quantifizierbarkeit von Prädikaten|Lambda-Ausdrücke|EinEindeutigkeitsquantor]
- Was gilt dann nicht mehr?** [Die Semi-Entscheidbarkeit von Allgemeingültigen Formeln]
 - Was ist ein Lambda Ausdruck?** [z.B. Definition eines neuen 2-stelligen Prädikates]
84. **Welche Typen von Agenten kennen Sie?** [Logikbasierte: Aktive Aktionen wie turn(right, ..., shoot, grab, release etc. | Reflex-Agenten: Reaktive, reagiert nur auf Perzepte der Umwelt wie z.B. „stench“, „breeze“ oder „glitter“| Modellbasierte Agenten: Haben ein interne Modell in dem die wesentliche Aspekte ihrer Umgebung, die Ausführbarkeit und Wirkung von Aktionen und weiteren Gesetzen der Welt und vor allem die eigenen Ziele gespeichert sind → Wichtigster Aspekt: Wie ändert sich die Welt? → Situationskalkül von McCarthy 1963]
85. **Was besagt das Situationskalkül von McCarthy?** [Es ist eine Weg, um mit PL1 dynamische Welten zu beschreiben, d.h. Zustände werden dann durch Terme repräsentiert → Ein Welt-Zustand kann nur durch Ausführen einer Aktion gewechselt werden → Eine Aktion hat Vorbedingungen und wird durch ihre Effekte beschrieben]
- Was sind Fluents?** [Fluents sind Relationen, die ihren Wahrheitswert über die Zeit ändern. Darstellung durch zusätzliches Argument, das ein Zustandsterm ist]
86. **Was versteht man unter positiven- bzw. negativen Effektaxiomen?** [Pos: Wenn man eine Aktion machen kann dann soll Sie auch gemacht werden! | Neg: Wenn man eine Aktion nicht machen kann, dann soll man es lassen ;-)]
87. **Was besagt das Rahmenproblem?** [Das Problem, wie man Merkmale beschreibt, die durch Operatoren nicht verändert werden d.h. wie sag ich dem Agenten der gerade Gold trägt, dass er dieses immer noch trägt obwohl er nach rechts gegangen ist, bzw dass er es nicht mehr trägt, wenn er ein „release“ erhalten hat|Man muss also den Zusammenhang zwischen zwei Merkmalen herstellen können, dies geschieht dann durch die Rahmenaxiome]
- Wie erhalte ich diese Axiome?** [Man muss sich diese für jedes einzelne Paar von Merkmalen und Aktionen konstruieren, dieses wird u.U. sehr aufwendig]
 - Es gibt auch solche Nachfolgezustandsaxiome, für was sind die?** [Sie sind eine etwas elegantere Art das Rahmenproblem zu lösen, denn Sie beschreiben den Nachfolgezustand einer Aktion vollständig]

88. **Diese Aktionsbeschreibung ist doch recht begrenzt, welche Handlungsgrenzen ergeben sich daraus?** [Zeit wird nicht explizit angegeben, das heisst wir können nicht darüber reden, wie lange eine Aktion dauert| Es ist nur ein agent vorgesehen, wünschenswert wären aber u.U. mehrere| Aktionen können nicht parallel ausgeführt werden| Keine kontinuierlichen Aktionen, wie das schieben eines Objektes von a nach b| Die Welt ist um den Agenten abgeschlossen, d.h. nur dieser kann sie verändern| Aktionen sind Deterministisch, d.h. sie werden mit absoluter Sicherheit ausgeführt→ Diese Beschreibungen sind dennoch für viele Fälle ausreichend]
89. **Was versteht man unter diagnostischen Regeln?** [Agentensicht: Regeln die aus einem Perzept direkt dessen Eigenschaften ableiten (z.B. Breeze→ Breezy)]
90. **Was sind dann aber kausale Regeln?** [Weltsicht: Kausale Regeln erklären die Eigenschaften der Welt bzw. wie man ein bestimmtes Perzept dann zu interpretieren hat| Bsp.: Um den Wumpus herum stinkt es→ wenn ich stinken wahrnehme, ist der Wumpus in einem der Nachbarfelder]
91. **Wie könnte man zwischen mehreren möglichen Aktionen auswählen?** [Bewerten z.B. nach Wünschbarkeitsklassen]
92. **Was versteht man unter Handlungsplanung?** [Man hat eine initiale Situation und eine gewünschte Zielbedingung. In einem Fundus möglicher Aktionen gilt es eine Sequenz möglicher Aktionen zu finden, die den Agenten vom Anfangs zum Zielzustand bringen]
93. **Was ist der Unterschied zwischen Handlungsplanung und Problemlösen?** [Bei der Handlungsplanung haben wir eine explizite, logikbasierte Repräsentation, beim Problemlösen nicht]
- Wir werden Zustände und Situationen jeweils beschrieben?** [HP: logische Formeln die die Weltzustände beschreiben| PL: Datenstrukturen]
 - Wie sieht es mit dem erkennen des Zieles aus?** [HP: Die Zielbedingung ist als logische Formel formuliert| PL: Es existiert ein Zieltest, der überprüft ob das Ziel erreicht ist (Black Box)]
 - Wie ist das mit den Operatoren?** [HP: Axiome oder Transformation von Formeln|PL: Modifikation von Datenstrukturen durch Programme]
94. **Was ist die Voraussetzung mit der Handlungsplanung auch tatsächlich eine Sequenz zum Ziel zu finden?** [Wenn der Initialzustand, alle Vorbedingungen, alle Nachfolgeaxiome und natürlich die Zielbedingung (was der Anfrage entspricht) vorhanden sind, dann findet man immer eine solche Sequenz!]
95. **Was ist der Strips Formalismus (STRIPS=Standard Research Institute Problem Solver)?** [Standardisierte Beschreibung des Weltzustands inkl. des Initialzustandes: Eine Menge von Grundatomen, keine Funktionssymbole ausser Konstanten→ Es ist immer nur der aktuelle Weltzustand zugreifbar (keine explizite Zustandsvariable)| Standardisierte Beschreibung der Zielbedingung als eine Menge von Grundatomen|Standardisierte Strips Operatoren]
- Wie sehen die Strips Operatoren aus?** [Die sind immer ein Tripel, bestehend aus dem Aktionsnamen, den Vorbedingungen und dem erzielten Effekt]
 - Was können wir nun damit konstruieren um effektiv eine Lösung zu suchen?** [Wir können uns einen Zustandsraum aufbauen, indem wir dann direkt suchen können und nicht mehr planen müssen. Der Zustandsraum hat dann im Allgemeinen einen niedrigen Verzweigungsfaktor und ist deshalb effizient zu durchsuchen. Die Suche geht sowohl vorwärts (Progressionsplanung) als auch rückwärts (Regressionsplanung)]
96. **Was versteht man unter Suchen im Planraum?** [Man hat am Anfang nur den Start und Zielzustand, mit Hilfe von Verfeinerungs- und Modifikationsoperatoren kann man nun den Plan konkreter machen bzw. ihn verändern.]
97. **Was versteht man unter einem nichtlinearen (partiell geordneten) Plan?** [Unter Umständen ist es bei einigen Problemen egal in welcher Reihenfolge man die einzelnen Schritte abarbeitet, man verwendet deshalb einen nichtlinearen (partiell geordneten) Plan. Aus diesem kann man dann die zugehörigen vollgeordneten linearen Pläne auslesen]
98. **Aus was besteht eigentlich ein Plan?** [Menge von Planschritten mit partieller Ordnung|Menge von Variablenbelegungen| Menge kausaler Beziehungen]
99. **Wann ist ein Planungsproblem vollständig?** [d.h. Jede Vorbedingung eines Schrittes wird erfüllt. zB Socken müssen vor Schuhen angezogen werden]
100. **Wann ist ein Planungsproblem konsistent?** [Ein Planungsproblem ist konsistent, wenn zwei Aktionen zwingend in bestimmter Reihenfolge nacheinander ausgeführt werden müssen, dann darf dies nicht umgekehrt erlaubt sein (also zB Socken über Schuhe ziehen)]

- Eindeutige Namen müssen vergeben werden, d.h. wenn $x = A$ ist dann darf x nicht auch B sein wenn A und B verschiedene Variablen sind]
101. **Wann hat man die Lösung für ein Planungsproblem?** [Wenn man einen sowohl vollständigen als auch konsistenten Plan hat]
 102. **Wenn wir unseren Plan sukzessive verfeinern, kann es passieren, dass kausale Beziehungen bedroht werden, dass heisst wir sind nach einem Schritt in einem inkonsistenten Zustand, nennen Sie ein Beispiel für ein solche Situation?** [Wenn wir aus dem partiellgeordneten plan einfach die verschiedenen seriellen möglichkeiten auslesen, kannes zB vorkommen bei dem Hardwarestore/Supermarkt Beispiel, dass wir eine Lösung erhalten die vorsieht Bananen im HWS zu kaufen]
 103. **Wie kann man diese Kausalitätsbedrohungen auflösen?** [Demotion: Den „Bedroher“ vor die kausale Beziehung legen |Promotion: Den Bedroher hinter die kausale Beziehung legen]
 104. **Was macht der POP Algorithmus?** [Er liefert einen Partiiell-geordneten-Plan (Partial Ordered Plan)]
 - a. Nennen Sie Eigenschaften des POP Algorithmus! [Er baut auf der Strips-Sprache auf; Er ist korrekt, das heisst jedes Resultat ist ein vollständiger und korrekter Plan|Er ist vollständig, d.h. falls IDS oder Breitensuche angewandt wird, findet er eine Lösung, falls eine existiert| Systematizität: Zwei verschiedene partielle Pläne haben nicht die gleichen total geordneten Pläne als Verfeinerungen, falls die partiellen Pläne keine Verfeinerungen voneinander sind (und total geordnete kausale Beziehungen enthalten)|Er sucht vom Ziel zum Ausgangszustand nach einem geeigneten Plan im Raum der partiellen Pläne]
 105. **Was passiert, wenn eine Variable in einem Literal eines „Effects“ auftaucht?** [Potentielle Bedrohung für eine kausale Beziehung]
 - a. **Wie kann man das lösen?** [Gleichheitsconstraint, Ungleichheitsconstraint| Erst durchführen, wenn Variable instantiiert ist→schwieriger festzustellen, ob ein Plan eine Lösung ist]
 106. **Welche Schritte muss man bei der Modellierung von Aufgaben durchführen?** [Entscheiden worüber man sprechen will|Ein Vokabular für Bedingungen, Operatoren und Objekte festlegen|Operatoren kodieren|Probleminstanzen kodieren]
 107. **Erklären Sie diese an Hand der Blockwelt?** [*Entscheiden worüber man sprechen will:* Das Problem ist es benannte Blöcke, die auf dem Tisch liegen in eine bestimmte Reihenfolge aufeinander zu stapeln| Auf einem Block kann aber nur ein anderer stehen|Bewegung ist nur möglich, wenn der Block frei ist, das heisst kein anderer auf ihm steht|Umstürzen ist nicht erlaubt|*Festlegen des Vokabulars:* Objekte a, b, c (Blöcke)|Prädikate $On(x, y) \rightarrow x$ liegt auf y ; $OnTable(x) \rightarrow x$ liegt auf der Tischplatte; $Clear(x) \rightarrow$ Es liegt nichts auf x |*Operatoren:* $move(x, y, z); stack(x, y), unstack(x, y) \dots$]
 108. **Wie ist Interferenz zwischen zwei Aktionen definiert?** [Zwei Aktionen interferieren gdw. die eine einen Effekt oder eine Vorbedingung der anderen löscht]
 - a. Nennen Sie ein Beispiel einer solchen Interferenz! [$geh(bn, fr), geh(bn, hh)$]
 109. **Wie ist ein Planungsgraph definiert?** [Ein Planungsgraph $\Pi(N, E)$ zu einem gegebenen Planungsproblem $P(D, O, I, G)$ ist ein gerichteter, in alternierende Ebenen geteilter Graph mit zwei Arten von Knoten. (Aktionsknoten A und Faktenknoten F) bzw zwei Arten von Kanten (Vorbedingungskanten FxA und Effektkanten AxF →Jede Ebene enthält nur eine Art von Knoten. Der Graph beginnt und endet mit einer Faktenebene. Kanten verbinden nur Knoten benachbarter Ebenen]
 110. **Wann ist eine Aktion auf Ebene Q anwendbar?** [Eine Aktion ist auf Ebene Q_0 anwendbar gdw. ihre Vorbedingungen in Q_0 enthalten sind]
 111. **Wann sind zwei Fakten p, q exklusiv voneinander auf Ebene $Q_{i \geq 1}$?** [Zwei Fakten p, q sind exklusiv voneinander auf Ebene $Q_{i \geq 1}$ gdw. es kein Paar nicht exklusiver Aktionen s, t auf Ebene S_{i-1} (oder keine einzelne Aktion gibt), die p und q als Effekt haben]
 112. **Wann haben zwei Aktionen s, t konkurrierende Vorbedingungen?** [Zwei Aktionen s, t haben konkurrierende Vorbedingungen, gdw. es Fakten $p \in pre(s)$ und $q \in pre(t)$ gibt, die exklusiv sind]
 113. **Wann sind zwei Aktionen s, t exklusiv voneinander?** [Zwei Aktionen s, t sind exklusiv voneinander ,auf Ebene S_0 , gdw. sie interferieren, auf Ebene $S_{i \geq 1}$ gdw. sie interferieren oder konkurrierende Vorbedingungen haben]
 114. **Wann sagt man, ein Planungsgraph habe einen Fixpunkt?** [Ein Planungsgraph hat einen Fixpunkt erreicht gdw. 2 Faktenebenen und alle ihre Exklusivitätsbeziehungen identisch sind]

115. **Wann ist das Planungsproblem bzgl. Des Fixpunktes unlösbar?** [Wenn ein Teilziel nicht in der letzten Faktenebene enthalten ist, oder 2 Teilziele exklusiv voneinander sind]
116. **Wie funktioniert Rückwärtssuche?** [Man startet auf der maximalen Faktenebene. Die Ziele auf dieser Ebene sind gegeben durch unser ursprüngliches Ziel. Anschließend wählt man einen Punkt (Auswahlpunkt) der eine minimale Menge von nicht-exklusiven Operatoren auf Ebene i die das Ziel auf Ebene $i+1$ erreichen] Solange der Fixpunkt noch nicht erreicht ist, wird eine weitere Ebene erzeugt und die Suche erneut gestartet (iterativ)
- Was heißt minimal?** [Minimal heißt, dass jeder Operator mindestens ein Teilziel erreicht, das von keinem anderen Operator erreicht wird → Die Vorbedingungen der Operatoren bilden die Ziele auf der Ebene i]
117. **Kann es sein, dass die Ziele nicht-exklusiv sind, aber trotzdem kein Plan gefunden werden kann?** [Ja! Nämlich dann, wenn auf einer Zwischenebene ein unmöglicher Zustand durchquert werden müsste]
118. **Sei i die Ebene, auf der der Graph seinen Fixpunkt erreicht hat. Sei $|G_i^t|$ die Anzahl aller unlösbaren Ziele, die der Planer auf i für einen Graphen der Tiefe t erzeugt hat. Wann besitzt ein Planungsproblem demnach keine Lösung?** [gdw. $|G_i^t| = |G_i^{t+1}|$ Diese Bedingung ist notwendig aber auch hinreichend]
119. **Leider ist unser Wissen über die Welt nicht immer vollständig. Aber, wir müssen trotzdem agieren und unseren Schluss unter Unsicherheit ziehen. Welche Möglichkeiten gibt es hierbei?** [Nicht-monotones Schließen|Schließen über Eintrittswahrscheinlichkeiten|Entscheidungen treffen welche den Nutzen maximiert und die Kosten minimiert]
120. **Welches Problem erhalten wir, wenn wir versuchen die ganze Welt in Regeln zu fassen?** [Geht nicht, da nicht zu jedem Symptom alle Ursachen bekannt sind. Kausale Regeln sind auch nicht anwendbar, da die Ursache u.U. nicht immer die gleichen Symptome hat (z.B. Cavity)]
121. **Warum können wir selbst (scheinbar) offensichtliche Fakten nicht als sicher betrachten?** [Die Sensoren können ungenau sein (z.B. beim Roboter); die Faktenwahrscheinlichkeit wird durch den Sensor nur erhöht oder erniedrigt, aber nicht sicher gestellt]
122. **Agenten sind von Regeln und Fakten nur bis zu einem gewissen Grad überzeugt! Wie kann man diesen ausdrücken?** [Durch Wahrscheinlichkeiten z.B. Der Agent glaubt seinem Sensor nur zu 0.9]
123. **Rationale Entscheidungen unter Unsicherheit treffen, ist das überhaupt möglich?** [Die verschiedenen Aktionen und Pläne die wir zur Auswahl haben können zu verschiedenen Ergebnissen führen mit verschiedenen Wahrscheinlichkeiten. Die Aktionen verursachen verschiedene subjektive Kosten, die Ergebnisse haben verschiedenen subjektiven Nutzen → Rational ist es also den Gesamtnutzen zu maximieren] Unsere Entscheidungstheorie ergibt sich also aus der Summe der Nutzentheorie und der Wahrscheinlichkeitstheorie (Entscheidungstheorie=Nutzentheorie+Wahrscheinlichkeitstheorie)]
124. **Was versteht man unter unbedingter bzw. a priori Wahrscheinlichkeit?** [Die Wahrscheinlichkeit, dass ein Ereignis A eintritt, im Falle das keine zusätzlichen Informationen verfügbar sind. Z.B. $P(\text{Karies})=0.1$]
125. **Was versteht man unter dem Wahrscheinlichkeitsvektor?** [Der Vektor der Wahrscheinlichkeiten bezeichnet den geordneten Wertebereich der Zufallsvariablen X . z.B. kann das Wetter unterschiedliche Wahrscheinlichkeiten für unterschiedliche Wetterzustände (sonnig, regnerisch, bewölkt etc.) haben]
126. **Was versteht man unter bedingter bzw. a posteriori Wahrscheinlichkeit?** [Die Wahrscheinlichkeit dass ein Ereignis A unter der Zusatzbedingung B eintritt geschrieben wird $P(A|B)$ → Bei Wahrscheinlichkeitsvektoren erhalten wir eine Tabelle von bedingten Wahrscheinlichkeiten aller Kombinationen der Werte einer Zufallsvariablen]
127. **Wie kann man die bedingte Wahrscheinlichkeit $P(A|B)$ unter Zuhilfenahme der zugehörigen unbedingten Wahrscheinlichkeiten berechnen, unter der Voraussetzung das $P(B)>0$ ist?** [$P(A|B)=P(A \wedge B)/P(B)$]
128. **Was bedeutet der Ausdruck $P(X,Y)$?** [$P(X,Y)=P(X|Y)P(Y)$. Entspricht dann einem Gleichungssystem → X und Y können Vektoren sein!]
129. **Was besagt die Produktregel?** [$P(A \wedge B)= P(A|B)P(B)=P(B|A)P(A)$]
130. **Wann sind A und B unabhängig voneinander?** [Falls $P(A|B)=P(A)$ und $P(B|A)=P(B)$]
131. **Wenn A und B unabhängig voneinander sind, gilt ein Spezialfall der Produktregel, welcher?** [$P(A \wedge B)=P(A) P(B)$]

132. **Nennen sie die 4 Axiome der Wahrscheinlichkeitstheorie!** [1.Axiom: $0 \leq P(A) \leq 1$] 2. Axiom: $P(\text{True})=1$] 3.Axiom: $P(\text{False})=0$] 4.Axiom: $P(A \vee B)=P(A)+P(B)-P(A \wedge B) \rightarrow$ Alle anderen Eigenschaften lassen sich aus diesen Axiomen ableiten]
133. **Wieso sind die Axiome sinnvoll?** [Wenn P eine objektiv beobachtbare Wahrscheinlichkeit bezeichnet, machen die Axiome Sinn, der Agent modelliert seine Überzeugung aber selber, sie ist somit subjektiv also ist die Frage gegeben warum er dann überhaupt auf die Axiome zurückgreifen sollte? Er sollte! Wie de Finetti bereits 1931 nachgewiesen hat]
134. **Wie lautet de Finettis Theorem von 1931?** [Ein Agent, der aufgrund von Wahrscheinlichkeiten wettet, die die Axiome nicht erfüllen, kann zu Wetten gebracht werden, in denen er sicher Geld verliert, ganz gleich welcher Fall wirklich eintritt]
135. **Erklären sie das Wettmodell von de Finetti?** [Hier wird auf nichtbeachten des 4. Axioms gesetzt ($P(A \vee B)=P(A)+P(B)-P(A \wedge B)$)] Nehmen wir an der Agent hat die Überzeugungen $P(A)=0.4$ und $P(B)=0.3$ und $P(A \vee B)=0.8$ und $P(A \wedge B)=0.0 \rightarrow$ Er kennt (beachtet) aber wohl das 4. Axiom nicht, nach welchem das Ergebnis von $P(A \vee B)=0.7$ wäre \rightarrow Der Agent verliert deshalb immer]
136. **Was sind Verbundwahrscheinlichkeiten?** [Die Disjunktion von atomaren Ereignissen, z.B. bei 2 Variablen erhalten wir die folgenden 4 atomaren Ereignisse: $X \wedge Y$, $X \wedge \neg Y$, $\neg X \wedge Y$, $\neg X \wedge \neg Y$]
137. **Was ist die Verbundwahrscheinlichkeitsverteilung?** [Weißt jedem atomaren Ereignis eines Wahrscheinlichkeit zu bei X und Y haetten wir zum Beispiel eine 2x2 Tabelle mit 4 Kombinationsmöglichkeiten von X und Y \rightarrow Die Summe aller Werte der Tabelle ist 1]
138. **Was ist der Vorteil von Verbundwahrscheinlichkeiten?** [Durch die atomaren Ereignisse lassen sich sehr einfach alle anderen interessanten Wahrscheinlichkeiten errechnen. Wir müssen einfach die relevanten atomaren Ereignisse aufsummieren, also als Disjunktion formulieren!]
139. **Wie erhält man sehr einfach die zugehörigen unbedingten Wahrscheinlichkeiten aus der Tabelle der Verbundwahrscheinlichkeiten ablesen?** [Zeilen- bzw. Spaltenweise aufsummieren]
140. **Welches Problem haben wir mir der Verbundwahrscheinlichkeit?** [Es entstehen sehr viele atomare Ereignisse (k^n bei n Zufallsvariablen mit k Werten), das heisst es ist schwierig darzustellen (nur in tabellarische Form) und alle Werte sind aufwendig zu ermitteln]
141. **Gibt es dichtere Darstellungen für Verbundwahrscheinlichkeiten bzw. gibt es eine effiziente Methode, diese Darstellung dann zu verarbeiten?** [Im allgemeinen nicht, aber moderne Systeme arbeiten direkt mit Verbundwahrscheinlichkeiten und versuchen die Rechnungen zu vereinfachen indem sie Annahmen über die Unabhängigkeit von Variablen machen]
142. **Was besagt die Bayessche Regel und wie leitet man Sie her?** [Sie ermöglicht es unbekannte Wahrscheinlichkeiten aus bekannten zu berechnen] Produktregel $P(A \wedge B)=P(A|B)P(B)$ bzw. $P(A \wedge B)=P(B|A)P(A) \rightarrow$ Wir setzen die beiden Terme gleich und schieben $P(B)$ auf die andere Seite und erhalten die Bayessche Regel||Man kann damit einfach die Wahrscheinlichkeit kausaler Beziehungen berechnen]
143. **Wie kann man mit der Bayesschen Regel auch sehr einfach relative Wahrscheinlichkeiten berechnen?** [Möchte man z.B $P(C|T)$ und $P(G|T)$ in Relation zueinander setzen, dann wendet man einfach die Bayessche Regel auf beide an, einmal auf den Wert über dem Bruchstrich und einmal als Kehrwert der Bayesschen Regel auf den Wert unter dem Bruchstrich. Es kürzt sich dann erfreulicherweise der Wert für $P(T)$ heraus, weshalb wir diesen vorab nicht schätzen müssen]
144. **Was ist Normalisierung?** [Die Ergebnisse der relativen Wahrscheinlichkeit auf 1 normieren um absolute Wahrscheinlichkeiten zu erhalten. Z.B. anstatt die absolute Wahrscheinlichkeit von $P(C|T)$ unter zu Hilfenahme von $P(T)$ (welches wir aber eigentlich nicht haben) direkt zu berechnen, kann man auch eine vollständige Fallanalyse durchführen und den Zusammenhang das $P(C|T)+P(\neg C|T)=1$ ist ausnutzen]
- a. **Geht das auch für mehrwertige Zufallsvariablen?** [Ja, unter Zuhilfenahme einer Normalisierungskonstante α , welche die Werte in $P(Y|X)$ zu 1 aufsummieren lässt]
145. **Warum sollte man trotz eines zu 99% richtigen Testergebnisses einer sehr seltenen Krankheit trotzdem nicht gleich den Mut verlieren?** [Wenn die Testgenauigkeit viel größer als die Häufigkeit der Krankheit ist, dann ist selbst das positive Ergebnis nicht so bedrohlich wie es scheint, denn der Test hat z.B. eine Fehlerwahrscheinlichkeit von 1%. Die Krankheit tritt aber nur in einem von 10000 Fällen auf]
146. **Was sind multiple Evidenzen und wie geht man geschickt mit ihnen um?** [Multiple evidenzen sind mehrere Bedingungen für eine Wahrscheinlichkeitsdiagnose, es wäre aber

ungeschickt, gleich mit allen Bedingungen gleichzeitig zu rechnen, deshalb nimmt man diese sukzessive hinzu. Dazu verwendet man eine Variante der Bayesschen Regel die sogenannte Bayessche Evidenzregel.

147. **Was ist ein Bayessches Netzwerk?** [Unsere Zufallsvariablen werden als Knoten dargestellt – Gerichtete Kanten zwischen zwei Zufallsvariablen symbolisieren direkten Einfluss – Jeder Knoten ist mit einer Tabelle bedingter Wahrscheinlichkeiten assoziiert, die den Effekt der Eltern auf den Knoten quantifiziert – Der Graph ist gerichtet und azyklisch]
148. **Was bedeutet Inferenz in Bezug auf die Bayesschen Netze?** [Berechnung der Verteilung einer Menge von Variablen → Die Komplexität der Inferenz hängt von der Struktur des Netzwerkes ab | Die Inferenz ist in bayesschen Netzen im Allgemeinen NP-vollständig im Gegensatz zu Polytrees, welche polynomiell in der Größe des Netzwerkes sind]
149. **Welche Eigenschaften haben die Bayesschen Netze?** [Sie sind eine kompakte Repräsentation für Verbundwahrscheinlichkeiten, erreicht durch Unabhängigkeitsannahmen | Unterstützen verschiedene Formen des Schließens bei gegebene Evidenzen: Kausal, Diagnostisch, Interkausal und Gemischt]
150. **Nennen Sie einige Alternative zu bayesschen Netzen!** [Nicht-monotone-Logik (Qualitative Variante) | Regelbasierte Systeme | Dempster-Shafer Theorie (mgl. Repräsentation von Ignoranz) | Fuzzy Logic und Fuzzy Mengen (Verarbeitung von Vagheit (also nicht Unsicherheit) „Das Auto fährt schnell“)]
151. **Was besagt die Nutzentheorie?** [Es gibt eine sogenannte Nutzenfunktion, die Zustände bewertet und formalisiert, so dass Bevorzugen bestimmter Zustände durch den Agenten | Je nach dem welche Aktion man ausführt, hat man | eine gewisse Wahrscheinlichkeit auch den gewünschten Folgezustand zu erreichen, was einem einen gewissen erwarteten Nutzen erbringt]
152. **Wie wird der erwartete Nutzen berechnet?**
 $[EU(A|E) = \sum_i P(\text{Result}_i(A)|\text{Do}(A), E) \times U(\text{Result}_i(A)) \rightarrow \text{Dieser Nutzen soll zum MEU (max expect. utility) maximiert werden}]$
153. **Welche Probleme ergeben sich beim MEU-Prinzip?** [$P(\text{Result}_i(A)|\text{Do}(A), E)$ erfordert ein vollständiges kausales Modell der Welt | Man muss permanent an Veränderungen anpassen | NP-Vollständig für Bayessche Netze | $(\text{Result}_i(A))$ erfordert Suche oder Planen. Um einen Zustand zu bewerten muss man die möglichen Folgezustände kennen, d.h. die „Wirkung des Zustandes auf die Zukunft“]
154. **Nennen Sie die folgenden Axiome der Nutzentheorie mit den gegebenen Zuständen A, B und C:**
- Orderability?** [Ein Agent muss wissen was er will, das heißt er muss sich zwischen zwei Zuständen entscheiden oder indifferent (unbestimmt) zwischen beiden sein]
 - Transitivität?** [Verletzung verursacht irrationales Verhalten ($A \succ B \succ C \succ A$)]
 - Kontinuität?** [$A \succ B \succ C \Rightarrow \exists p [p, A ; 1 - p, C] \sim B$]
 - Substituierbarkeit?** [Einfachere Zustände können gegen komplizierter ersetzt werden ohne dass sich die Indifferenz verändert
 $A \sim B \Rightarrow [p, A ; 1 - p, C] \sim [p, B ; 1 - p, C]$]
 - Monotonie?** [Bevorzugt ein Agent einen Nachfolgezustand A, so muss er die Aktion wählen, die diesen Zustand mit der größten Wahrscheinlichkeit erreicht
 $A \succ B \Rightarrow$
 $(p \geq q \Leftrightarrow [p, A ; 1 - p, B] \succeq [q, A ; 1 - q, B])$]
 - Dekomponierbarkeit?** [Regel zum „Ausmultiplizieren“ von Ausdrücken:
 $[p, A ; 1 - p, [q, B ; 1 - q, C]] \sim$
 $[p, A ; (1 - p)q, B ; (1 - p)(1 - q), C]$]

155. **Was ist das Utility principle?** [Beachtet ein Agent in seinen Präferenzen die Axiome, dann

$$U(A) > U(B) \Leftrightarrow A \succ B$$

existiert eine Funktion $U: S \rightarrow R$ mit: $U(A) = U(B) \Leftrightarrow A \sim B$

156. **Was ist das Maximum expected utility principle?**

$$U([p_1, S_1; \dots; p_n, S_n]) = \sum_i p_i \times U(S_i)$$

157. **Wie wird die Nutzenfunktion normiert?** [Werte sollten zwischen 0 und 1 liegen, d.h. bester Preis ist dann = 1 und der schlechteste Fall soll = 0 sein]

158. **Es gibt eine deterministische und eine stochastische Variante, Entscheidungen zu treffen. Wie ist das gemeint?** [Bei der deterministischen Variante tut der Agent immer das, was man ihm vorschreibt. Bei der stochastischen Variante befolgt er die Vorschrift nur zu einem bestimmten Prozentsatz, ansonsten läuft er woanders hin]

159. **Was versteht man unter dem Markov Entscheidungsproblem?** [Die Suche nach einer optimalen Strategie, die den erwarteten Nutzen maximiert]

160. **Welche Strategie zur Lösung des Markov Entscheidungsproblems (MDP) haben wir kennen gelernt?** [Value Iteration]

a. **Was ist die Grundidee dabei?** [Für jeden Zustand wird sein Nutzen berechnet. Ausgehend vom Nutzen kann dann eine optimale Aktion für jeden Zustand ausgewählt werden]

b. **Wann heisst eine Nutzenfunktion separierbar?** [Eine Aktionsfolge generiert eine Baum möglicher Zustände (Histories). Die Nutzenfunktion über diese Histories heisst nun separierbar, wenn es eine Funktion f gibt, die einen einzelnen Zustand s_0 von der Nutzenfunktion selbst separiert, also:

$$U_h([s_0, s_1, \dots, s_n]) = f(s_0, U_h([s_1, \dots, s_n]))$$

in der einfachsten Variante ist f eine Belohnungsfunktion die meinen einzelnen separierten Zustand in Abhängigkeit der möglichen Aktionen bewertet damit die Aktion mit dem maximalen Nutzen ausgewählt werden kann!]

c. **Wie funktioniert Value Iteration genau?** [Initialisiere eine Funktion $V(s)$ beliebig → Nehme einen Zustand s_i aus S und teste auf ihn alle möglichen Aktionen → Schreibe diese Tupel in eine Menge $Q(s,a)$ → Wenn alle Aktionen getestet wurde, dann wähle das maximum aus $Q(s,a)$ bezüglich a und Schreibe den wert in $V(s)$ → Nehme des nächste s_i und fahre iterativ fort → V ist am Ende die maximierte Aktionsfolge]

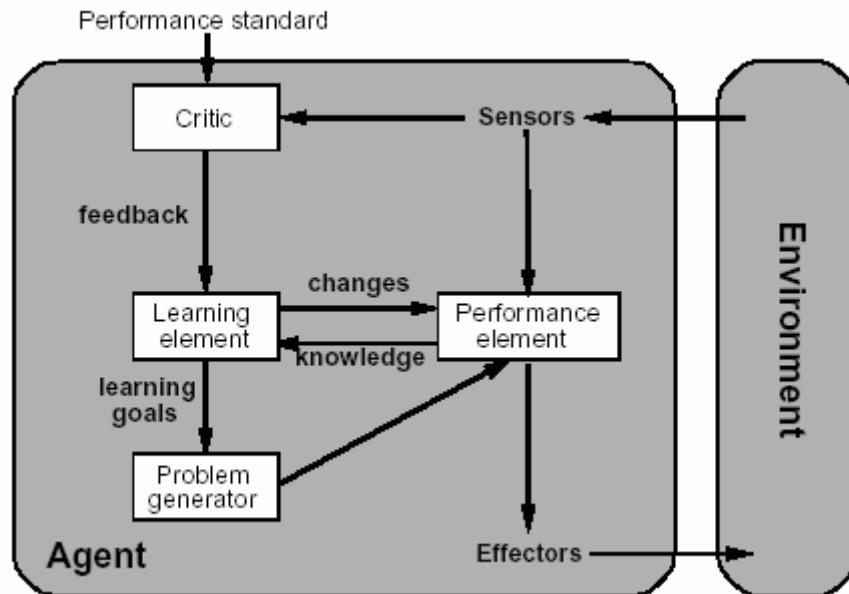
d. **Welche Verbesserungsmöglichkeit hat man, wenn der Nutzen der Endzustände bekannt ist?** [Man muss das Entscheidungsproblem dann nur für $n-1$ ausführen, der letzte Schritt ergibt sich automatisch]

e. **Welche Probleme können durch Zyklen im Entscheidungsprozess erzeugen?** [Die Historie ist potentiell unendlich lange. Lösung mit:

$$U_{t+1}(i) \leftarrow R(i) + \max_a \sum_j M_{ij}^a U_t(j)$$

161. **Was ist lernen?** [Ein Agent lernt, wenn er durch Erfahrung seine Fähigkeit, eine Aufgabe zu lösen, verbessern kann]

162. **Wie lernt ein Agent und welches sind die Bausteine?**



[Performance-Element: Verarbeitet Wahrnehmungen und wählt Aktionen aus (entspricht dem Agenten wie bisher)|Learning Element: Durchführen von Verbesserungen| Critic: Bewertung des Agentenverhaltens auf den Grundlagen eines gegebenen externen Verhaltensmassstabes| Problem-Generator: Vorschlägen von explorativen Aktionen, die den Agenten zu neuen Erfahrungen führen]

- a. **Welche Fragen beeinflussen das Learning Element?** [Welche teile des Performanceelements verbessert werden sollen|Welche Repräsentation gewählt wird| Welche Form von Rückkopplung verfügbar ist| Welche Ausgangsinformation für den Lernprozess zur Verfügung steht]
163. **Welche Arten der Rückkopplung kennen Sie?** [Ziel des Lernens ist das Annähern der tatsächlichen an die ideale Funktion dabei gibt es 3 verschiedene Rückkopplungstypen dem Agenten mitzuteilen wie weit er vom Optimum entfernt ist. Supervised Learning/Reinforcement Learning/Unsupervised Learning]
 - a. **Was ist Supervised Learning?** [E/A für Supervisor verfügbar. Ein Lehrer teilt dem System den Effekt auf die Umwelt und damit die korrekte Aktion mit]
 - b. **Was ist Reinforcement Learning?** [Je nach Erfolg der Aktion wird der Agent bestraft oder belohnt]
 - c. **Was ist Unsupervised Learning?** [Der Agent kann nur Modelle für das Auftreten von Regelmäßigkeiten seiner Beobachtungen Lernen, aber nicht was er richtigerweise tun müsste]
164. **Was versteht man unter induktivem Lernen?** [Aus gegebenen Werte z.B. $(x, f(x))$ Wertpaare soll man nun eine möglichst gute Hypothese finden, die die Funktion f möglichst gut approximiert. Man schließt also aus den gegebene Werte der Funktion f auf die gesamt Funktion f] Achtung: Problem BIAS!]
165. **Wie sieht denn so ein Entscheidungsbaum aus?** [Die Eingabe ist die Beschreibung einer Situation durch eine Menge ihrer Eigenschaften|Die Ausgabe ist jeweils JA bzw. NEIN bzgl. eines Zielprädikates]
166. **Und was ist das Ziel des Lernprozesses?** Definition eines Zielprädikates in Form eines Entscheidungsbaumes]
167. **Wie kann man dieses Zielprädikat Repräsentieren?** [Man kann den Entscheidungsbaum als eine Konjunktion von Implikationen schreiben, eine Implikation ist dann natürlich nur ein Pfad des Baumes der auf „YES“ endet]
168. **Kann man denn alle Formeln als Entscheidungsbaum darstellen?** [ja und nein, alle aussagenlogischen Formeln sind nach Theorem darstellbar, aber leider nicht alle Prädikatenlogischen, man kann die Entscheidungstests nicht auf mehrere Prädikatenlogische Objekte gleichzeitig beziehen→Es wäre aber möglich mehrere Objekte zu einem Test zusammenzufassen, aber der Entscheidungsbaum würde dann exponentiell wachsen!]

169. **Nennen Sie zwei Beispiele für Funktionen, die auf jeden Fall einen Baum exponentieller Größe erfordern!** [Parity Funktion|Majority Funktion| Es gibt also keine kompakte Repräsentation für alle Booleschen Funktionen]
170. **Was ist mit Ockham's Razor gemeint?** Wenn es für eine Begebenheit eine einfache und eine komplizierte Erklärung (Hypothese) gibt wählt man grundsätzlich die einfachere, zumindest solange wie diese Erklärung den Tatsachen standhält| Bsp: Die Welt wurde von Gott erschaffen vs. Das Universum hat seine Singularität im Urknall gelöst und expandiert seit dem unaufhaltsam...]
171. **Wie erhält man einen trivialen Entscheidungsbaum?** [In dem man jedes Beispiel (Tatsache) in einem Pfad des Baumes repräsentiert]
- Welche Eigenschaften hat ein solcher Baum?** [Er repräsentiert nur die Erfahrungen des Agenten| Es ist keine Extraktion allgemeinerer Muster möglich| Er besitzt keine Vorhersagekraft| Nach Ockhams Razor würde man eine minimalen Baum bevorzugen, leider ist das Erzeugen des kleinsten Entscheidungsbaums nicht handhabbar, deshalb muss man auf Heuristiken zurückgreifen]
 - Wie sollte man die Attribute gewichten, d.h. in welcher Reihenfolge man die Attribute abfragen sollte?** [Man wählt das Attribut aus, das die größtmögliche Unterscheidung der Beispiele erlaubt]
172. **Welche 4 Fälle müssen beim rekursiven Lernverfahren beachten?** [1. Wenn es noch positive oder negative Beispiele gibt, wähle neues Attribut| 2. Nur positive (oder nur negative) Beispiele: fertig| 3. keine Beispiele: es gab kein Beispiel mit dieser Eigenschaft. Antworte JA, wenn Mehrzahl der Beispiele des Vaterknotens positiv ist, sonst NEIN| 4. Keine Attribute mehr, aber noch Beispiele mit unterschiedlicher Klassifikation: es lagen Fehler in den Daten vor (→ NOISE) oder die Attribute sind unzureichend. Reagiere wie im vorherigen Fall]
173. **Was ist die Trainingsmenge?** [Teil der Gesamtmenge von Beispielen. Die Trainingsmenge wird benutzt um eine Hypothese H zu erstellen. Je größer die Trainingsmenge, desto besser ist die Vorhersagekraft]
174. **Was ist die Testmenge?** [Teil der Gesamtmenge von Beispielen disjunkt zur Trainingsmenge. Die Testmenge wird benutzt, um den Anteil korrekt klassifizierter Beispiele zu messen, d.h. die Qualität des Lernalgorithmus zu beurteilen]
175. **Was ist besonders wichtig bei der Behandlung von Test- bzw. Trainingsmenge?** [Die beiden Mengen müssen unbedingt getrennt gehalten werden, damit diese unabhängig bleiben. Ein beliebter Fehler in diesem Zusammenhang ist es, dass der Lernalgorithmus nach einer Modifikation auf beide Mengen getestet wird, leider wird dadurch aber wissen der Testmenge in den Algorithmus gesteckt. Die Unabhängigkeit zwischen Trainings und Testmenge ist dann weg]
176. **Wie funktioniert also nun die Bewertung des Lernalgorithmus?** [Zuallererst muss man eine große Anzahl von Beispiel zur Verfügung haben bzw. sammeln. Die Beispiele werden in zwei beliebige disjunkte Mengen aufgeteilt, die eine nennen wir Trainingsmenge die andere nennen wir Testmenge. Mit Hilfe der Trainingsmenge wird versucht die Hypothese zu modellieren und nach und nach zu verfeinern. Mit Hilfe der Testmenge wird überprüft wie gut die Hypothese modelliert wurde bzw. wie hoch der Anteil korrekt klassifizierte Beispiele ist. Das Verfahren wird für eine andere zufällig gewählte Trainings- und Testmenge wiederholt, es darf aber nicht passieren, dass Elemente der vorherigen Trainingsmenge bei einem weiteren Durchlauf plötzlich in der Testmenge stehen, da die Mengen sonst nicht mehr unabhängig wären.]
177. **Wie bewerten man ein Attribut?** [Die Frage ist immer wie viel Vorabinformation man über den Ausgang des Attributes hat. Wenn man z.B. weiss, dass eine Münze gezinkt ist, sodass Sie häufiger auf eine bestimmte Seite fällt, dann ist der Wert dieser Vorabinformation natürlich sehr hoch. Der wert des Ausgang dabei ist sehr gering, da man das Ergebnis ja im Prinzip schon weiss.]
178. **Wie bestimmt man den Wert einer Information „in Bit“?** [Man nimmt die Ereigniswahrscheinlichkeiten und multipliziert diese jeweils mit dem dualen Logarithmus ihres Kehrwertes. Diese Ergebnisse werde aufsummiert, also:

$$I(P(v_1), \dots, P(v_n)) = \sum_{i=1}^n P(i) \text{ld}\left(\frac{1}{P(i)}\right)$$

179. **Wie hoch ist der durchschnittliche Informationsgehalt nach Auswahl von A?**

$$R(A) = \sum_{i=1}^v \frac{p_i + n_i}{p + n} \times I \left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i} \right)$$

[
180. **Wie berechnet sich der Informationsgewinn (Gain) durch die Auswahl von A?**

$$\text{Gain}(A) = I \left(\frac{p}{p + n}, \frac{n}{p + n} \right) - R(A)$$

[
→ Bei Gleichverteilung der Ereignisse, ist der vordere Teil „1“]

181. **Was versteht man unter Noise (Rauschen)?** [Das sind zufällige Fehler in den Lerndaten → Größere Bäume machen mehr Fehler auf neuen Daten (Overfitting) → Vermeidung durch ein sogenanntes Validation Set, d.h. die Trainingsmenge wird in zwei Teilmengen aufgeteilt (70%/30%). Mit den 70% wird der Baum aufgebaut, die übrigen 30% werden verwendet um eine angemessene Größe des Baumes zu bestimmen (Pruning!)]
182. **Wann ist eine Hypothese H „false negative“ bzw. „false positive“?** [false negative bedeutet dass die Hypothese sagt, das Beispiel sei negativ, in Wirklichkeit ist es aber positiv] false positiv bedeutet dass die Hypothese sagt, das Beispiel sei positiv, in Wirklichkeit ist es aber negativ]
183. **Was versteht man beim „Current Best Hypothesis“ Verfahren unter Generalisierung bzw. Spezialisierung?** [Falls ein Beispiel false positiv ist, wird die Hypothese spezialisiert, das heisst zusätzliche Terme werden hinzugefügt um die Hypothes bzgl. Beispiels wieder richtig zu klassifizieren] Falls ein Beispiel false negativ ist, ist die Hypothese zu speziell, d.h. Terme der Hypothes werden gestrichen und diese damit generalisiert]
184. **Wie genau funktioniert nun das Verfahren „Current Best Hypothesis“?** [Das Erste Beispiel (einer Belegung) wird genommen und eine konsistente, wahre Hypothese zu diesem Beispiel aufgebaut. Wird nun ein zusätzliches Beispiel hinzugenommen. Kann es sein, dass die Hypothese entweder weiter konsistent bleibt, oder „false negativ“ bzw. „false positiv“ ist. Nach der Konsistenzwiederherstellung wird das nächste Beispiel getestet. Ist eine Konsistenzwiederherstellung nicht möglich, scheitert das Verfahren]
185. **Welche Nachteile hat das Verfahren „Current Best Hypothesis“?** [Alle vorhergehende Beispiel müssen bei jeder Iteration erneut geprüft wird] Es wird nicht immer die einfachste Hypothese gefunden] Gute heuristiken für die Suche sind schwer zu finden] Suche führt leicht in eine Sackgasse die eine konsistent Hypothes nicht mehr ermöglicht → Lösung: Backtracking aber Hypothesenraum riesig → Besser wird das mit Version Space Learning gelöst]
186. **Was ist das Prinzip des Version-Space/Candidate-Elimination Learning?** [Zunächst wird eine Menge aller Hypothesen aufgestellt, und anschließend werden iterativ diejenigen herausgestrichen, die inkonsistent mit den Beispielen sind]
187. **Der Hypothesenraum ist sehr groß. Wie kann der Hypothesenraum kompakt repräsentiert werden?** [Die Hypothesen werden nicht alle aufgezählt, sondern in einem sogenannten Boundary Set abgegrenzt]
188. **Was ist ein Boundary-Set?** [Das Boundary Set begrenzt den Hypothesenraum durch die allgemeinst möglichen Hypothesen und die speziellsten. Alle dazwischenliegenden Hypothesen sind in unserem Set Das Set der allgemeinsten Hypothesen bezeichnet man als G-Set und das der speziellsten als S-Set. Eine Hypothese H ist konsistent gdw. sie eine Generalisierung eines Elements von S-Set ist und eine Spezialisierung eines Elements von G-Set ist, also bzgl G-Set TRUE und bzgl S-Set FALSE]
189. **Wie wird das Boundary in den einzelnen Iterationsschritten ge-updated?** [Es gibt 4 Möglichkeiten wenn ein neues Beispiel X getestet werden soll:
i. X ist false positiv für S_i → S_i ist zu allgemein und wird gestrichen
ii. X ist false negativ für S_i → S_i ist zu spezifisch und muss generalisiert werden
iii. X ist false positiv für G_i → G_i ist zu allgemein und muss spezialisiert werden
iv. X ist false negativ für G_i → G_i ist zu spezifisch und wird gestrichen]
190. **Wann bzw. wie Terminiert der Algorithmus?** [Es gibt 3 „Endszenarien“: Es bleibt nur genau eine Hypothes übrig] Der Hypothesenraum kollabiert, das heisst S und/oder G ist leer → Eine konsistente Lösung ist dann nicht möglich] Eine Menge von möglichen Hypothesen bleibt übrig → Die Disjunktion ist dann das Ergebnis]

191. **Welche Probleme gibt es bei diesem Algorithmus?** [Rauschen oder mangelnde Information führt immer zum Kollabieren des Hypothesenraumes] Bei beliebiger Disjunktion enthält S eine speziellste- und G eine allgemeinste Hypothes: *ANMERKUNG: Das Problem haben wir irgendwie nicht kapiert, wo ist das Problem?*
192. **Wofür braucht man PAC-Lernen?** [Das Problem beim konstruieren einer guten Hypothese ist es zu wissen, wie nah man an der tatsächlichen Funktion f ist mit seiner Approximation, denn f kennt man ja gar nicht. Diesem Problem begegnet das PAC-Lernen]
193. **Wann heisst eine Hypothese approximativ korrekt?** [Wenn die Errorfunktion (error(h)) angewandt auf h immer kleiner oder gleich einer Konstanten ϵ ist. $\rightarrow \text{error}(h) \leq \epsilon$]
194. **Wie hoch ist dann die Wahrscheinlichkeit, dass wir eine falsche Hypothese aufgrund der Konsistenz mit den ersten m Beispielen als richtig erachten?** [Der Fehler liegt zwischen $1-\delta$ und ϵ , \rightarrow Delta ist die untere Schranke für die Fehlerwahrscheinlichkeit. Für Delta gilt: $|H|(1-\epsilon)^m < \delta \rightarrow$ d.h. die Anzahl der zu testenden Beispiele m muss exponentiell wachsen für die booleschen Funktionen, da H doppelt Exponentiell wächst (sehr uneffektiv zum Lernen). Wir müssten den Hypothesenraum eigentlich einschränken, aber dadurch könnte f eliminiert werden, d.h. aus dem Hypothesenraum H gestrichen werden.]
195. Wieviel Beispiele benötigen wir dann also um zwischen den Fehlerwahrscheinlichkeiten $1-\delta$

und ϵ zu liegen? [$m \geq \frac{1}{\epsilon} (\ln \frac{1}{\delta} + \ln |H|)$]

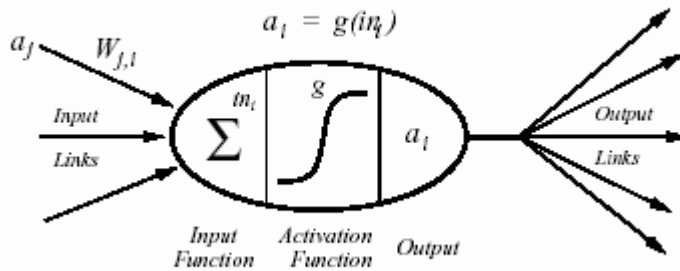
196. **Was ist Lernen von Entscheidungslisten?** [Wenn man beliebig lange Tests zulässt, dann kann man jede Boolesche Funktion repräsentieren. Werden nur Tests der Länge k zugelassen, dann kann der Lernalgorithmus erfolgreich von einer kleinen Menge von Beispielen lernen. Diese Sprache nennt man k-DL. Man kann leicht zeigen, daß die Sprache k-DL eine Obermenge der Sprache k-DT (Menge aller Entscheidungsbäume mit höchstens Pfadtiefe k) ist. Notation: k-DL(n) ist die Sprache k-DL mit n Booleschen Attributen. k-DL ist lernbar: d.h. Jede Funktion in k-DL kann approximiert werden, nachdem man eine ausreichend große Anzahl von Beispielen gesehen hat. Um das zu tun muß man die Menge der Hypothesen in der Sprache berechnen. die Zahl der Beispiele, um eine k-DL Funktion zu lernen (PAC-learning) ist polinomiell in n (n ist die Anzahl der Attribute). Entscheidungslisten sind demnach ein effektives Verfahren, um Vorhersagen zu treffen.]
197. **Was ist der Vorteil der Verwendung eines natürlichen neuronalen Netzes?** [Hohe Geschwindigkeit durch massive parallelverarbeitung] Hohe Fehlertoleranz selbst bei Ausfällen von Teilnetzen| Nur langsamer Funktionsabfall bei fortschreitenden Ausfällen von Neuronen (Graceful degradation)|Gut geeignet für induktives Lernen]
198. **Erklären Sie die folgenden Grundbegriffe eines künstlichen neuronalen Netzes!**
- Einheiten?** [stellen die Knoten (Neuronen) im Netz dar]
 - Verbindungen?** [Kante (Synapse) zwischen den Knoten des Netzes. Jeder Knoten hat Ein- und Ausgabekanten]
 - Gewichte?** [Jede Kante hat eine Gewichtung, in der Regel eine reelle Zahl]
 - Ein- Ausgabeknoten?**[Besonders gekennzeichnete Knoten die mit der Aussenwelt verbunden sind]
 - Aktivierungsniveau?** [Der von einem Knoten aus seinen Eingabekanten zu jedem Zeitpunkt berechnete Wert. Dieser Wert wird über Ausgabekanten an die Nachbarknoten weitergeleitet]
199. **Wie funktioniert so eine Einheit?** [Es wird die Stärke der Eingabe für die Einheit berechnet, als Linearkombination aus der Eingabeaktivierung und dem Gewicht der Kante über alle Knoten die mit der Einheit verbunden sind:

$$in_i = \sum_j W_{j,i} \times a_j = \mathbf{W}_i \cdot \mathbf{a}_i$$

. Die Aktivierungsfunktion berechnet mit Hilfe

einer i.allg. nicht linearen Funktion g das Aktivierungsniveau a_i :

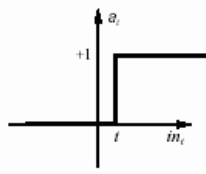
$$a_i := g(in_i) = g\left(\sum_j W_{j,i} \times a_j\right)$$



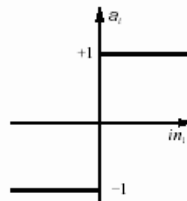
8

200. Nennen Sie Beispiele für Aktivierungsfunktionen!

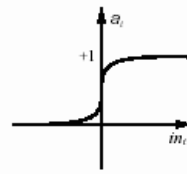
$$step_t(x) = \begin{cases} 1, & \text{if } x \geq t \\ 0, & \text{if } x < t \end{cases} \quad sign(x) = \begin{cases} +1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases} \quad sigmoid(x) = \frac{1}{1+e^{-x}}$$



(a) Step function



(b) Sign function



(c) Sigmoid function

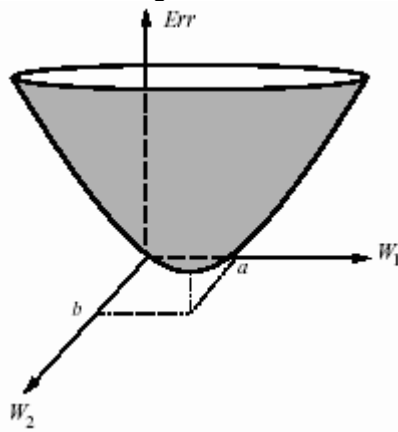
- [
201. **Was kann man mit neuronalen Netzen darstellen?** [z.B. Funktionen, insbesondere Boolesche Funktionen. Durch Verschachtelung von AND OR und NOT kann man jede beliebige boolesche Funktion darstellen. Netze werden dann allerdings sehr groß werden ($O(2^n/n)$)]
202. **Welche beiden Netzwerktopologieklassen kennen Sie?** [Rekurrent Netze: Verbindungen können beliebige Topologien aufweisen] Feed-Forward-Netze: Verbindungen stellen gerichteten azyklischen Graphen (DAG) dar]
203. **Was ist denn ein Feed-Forward Netz genau?** [Sie werden normalerweise in Schichten konstruiert. Dabei gibt es keine Verbindung zwischen Einheiten einer Schicht, sondern immer nur zur nächsten Schicht. FF Netze sind einfach zu verstehen, da ihre Ausgabe einfach eine Funktion der Eingaben und der Gewichte ist]
204. **Nennen sie einige Eigenschaften Rekurrenter Netze?** [Oft instabil, oszillierend, zeigen chaotische Verhalten, interner Zustand bildet sich durch Backpropagation im Aktivierungsniveau ab]
205. **Welche Typen rekurrenter Netze kennen Sie?** [Hopfield Netze|Boltzmann-Maschinen]
- Erläutern Sie Hopfield-Netze genauer!** [Rekurrentes Netz mit symmetrischen (Gewichte gleich), bidirektionalen Verbindungen] Alle Einheiten sind sowohl Eingabe als auch Ausgabeeinheiten| Aktivierungsfunktion ist sign| Realisierung eines assoziativen Speichers| Ein Hopfield Netz kann $0.138 \times N$ Beispiele sicher speichern, wobei N die Zahl der Einheiten im Netz ist]
 - Erläutern Sie die Boltzmann-Maschinen genauer!** [Symmetrische Gewicht, aber es gibt Einheiten die weder Eingabe noch Ausgabe-Einheiten sind (Zwischeneinheiten) | Stochastische Aktivierungsfunktion, d.h. die Zustandsübergänge ähneln Suchverfahren mit simulated annealing| Formal zur Variational Bayesscher Netz identisch, die mit stochastischer Simulation bearbeitet werden]
206. **Was ist ein geschichtetes Feed-Forward Netz?** [Es gibt Eingabeeinheiten, Ausgabeeinheiten und Hidden-Units ohne direkte Verbindung nach draussen]
207. **Welche Arten von Funktionen kann man mit einem GFF darstellen?** [Mit einem GFF mit einer verborgenen Schicht, kann man jede beliebige stetige Funktion darstellen. In einem GFF mit zwei verborgenen Schichten können zusätzlich diskontinuierliche Funktionen dargestellt werden]

208. **Was ist denn Lernen in einem neuronalen Netz?** [Das Anpassen der Parameter (Gewichte) durch die gegebenen Beispiele der gesuchten Funktion. Die Netzwerktopologie ist fest vorgegeben]
209. **Wie wählt man ein gutes Netz?** [Schweres, nicht durch eine Theorie gelöstes Problem. Denn ein optimales Netz ist u.U. exponentiell groß. Ist das Netz zu klein, kann die gewünschte Funktion u.U. nicht dargestellt werden, ist das Netz hingegen zu groß, dann lernt das Netz die Beispiele auswendig ohne zu generalisieren → Overfitting] Zur Lösung gibt es nur Heuristiken
210. **Was ist das Prinzip Optimal Brain Damage Heuristik?** [Starte mit einem Netz maximaler Verbindungen (Vollvermascht?) Füttern mit Beispielen|Nach dem 1. Training wird die Anzahl der Verbindungen reduziert mit Hilfe der Informationstheorie→ Iteriere diesen Prozess| Das Verfahren gibt es auch umgekehrt, also von kleinen Netzen zu einem größeren, besseren!]
211. **Was versteht man unter einem Perzeptron?** [Ein einschichtige FF-Netze, welche insbesondere in den 50er Jahren studiert wurden, da sie die einzigen GFF-Netze waren für die Lernverfahren bekannt waren.]
- Was können Perzeptron darstellen bzw. lernen?** [Genau die Klasse der linear separierbaren Funktionen, d.h. Teilen zwei Punktmengen im Raum durch eine Gerade bzw. Ebene im Raum]
 - Wie wird das Netzwerk am Anfang initialisiert?** [Mit zufälligen Gewichten zwischen -0.5 und 0.5]
 - Was besagt die Perzeptron-Lernregel?** [T ist die korrekte Ausgabe| O ist die tatsächliche Ausgabe| Err ist die Differenz aus T und O (T-O)| Ist Err positiv, so muss O erhöht werden, ist Err negativ, so muss O erniedrigt werden| Nach einem Satz von Rosenblatt(1960) konvergiert die Perzeptron-Lern-Regel immer zu korrekten Werten bei linear separierbaren Funktionen]
 - Wie beeinflusst man O?** [Jede Eingabeeinheit W_j trägt zu O bei.

$$W_j := W_j + \alpha \times I_j \times Err$$

α ist die Lernrate]

- Wieso funktioniert die Perzeptron-Lernregel?** [Bei linear separierbaren Funktionen gibt es keine lokalen Minima. Wir haben einen Gradientenabstieg, wobei jeder Gradient durch die Fehlerfläche bestimmt wird. Jede Gewichtungskonfiguration bestimmt einen Punkt auf der Fläche:



212. **Was können Sie zu mehrschichtigen FF-Netzen sagen?** [Sind geeignet zum Lernen des Restaurantproblems, aber Lernalgorithmen für FF-Netze sind nicht effizient und konvergieren nicht zum globalen Minimum. Vom PAC-Lernen her weiß man, dass Lernen allgemeiner Funktionen aus Beispielen im ungünstigsten Fall nicht handhabbar ist, unabhängig von der Methode]
213. **Was versteht man unter Backpropagation?** [Dabei wird versucht den Fehleranteil der Gewichte $W_{j,i}$ und $W_{k,j}$ zu bestimmen und entsprechend zu ändern. Der Fehler wird rückwärts durchs Netz propagiert]

$$W_{j,i} := W_{j,i} + \alpha \times a_j \times \underbrace{Err_i \times g'(in_i)}_{\Delta_i}$$

g muss eine ableitbare Funktion sein, weshalb als Aktivierungsfunktion gerne die Sigmoid-Funktion verwendet wird.

- a. **Um wie viel müssen wir nun die Gewichte ändern?** [Abhängig von Δ_i und W_{ji} :

$$\Delta_j = g'(in_j) \times \sum_i (W_{j,i} \times \Delta_i)$$

$$W_{k,j} := W_{k,j} + \alpha \times I_k \times \Delta_j$$

- b. **Warum funktioniert Back-Propagation?** [Wie bei Perzeptron Lernregel kann man auch die Back-Propagation als Gradientenabstieg auffassen, aber es existieren lokale Minima die man mit Hilfe von Neustarts, Rauschen und Tabusuche in den Griff bekommen muss]

214. **Nennen Sie ein paar Beispiel in denen neuronale Netze erfolgreich zur Anwendung kamen?** [Lernen der Aussprache: Programm Nettalk lernt nach 50 Epochen auch Text mit 1024 Wörtern 95% korrektheit auf der Trainingsmenge und 78% auf Testmenge] Erkennen handgeschriebener Postleitzahlen: Training auf 7300 Beispielen → Korrektheit 99%] Autofahren: Lernen durch Beobachten eines menschlichen Fahrers, nach 5 Minuten Testfahrt kann die Trainingsstrecke bereits mit 70mph befahren werden.]